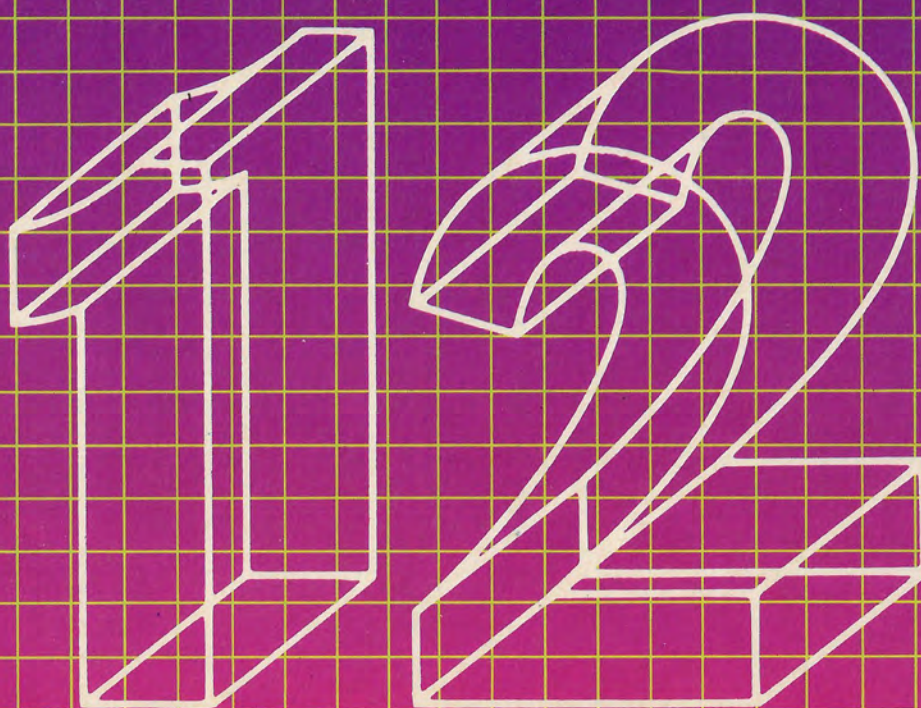


BIBLIOTECA PRACTICA

TRUCOS - SPRITES
BRICOLAGE DEL HARD
AULA ABIERTA
LOGO - TERMINOLOGIA

TALLER DE INFORMATICA

PROGRAMAS VALIDOS PARA AMSTRAD,
IBM, SPECTRUM, COMMODORE Y MSX



¿UN TALLER DE INFORMATICA?

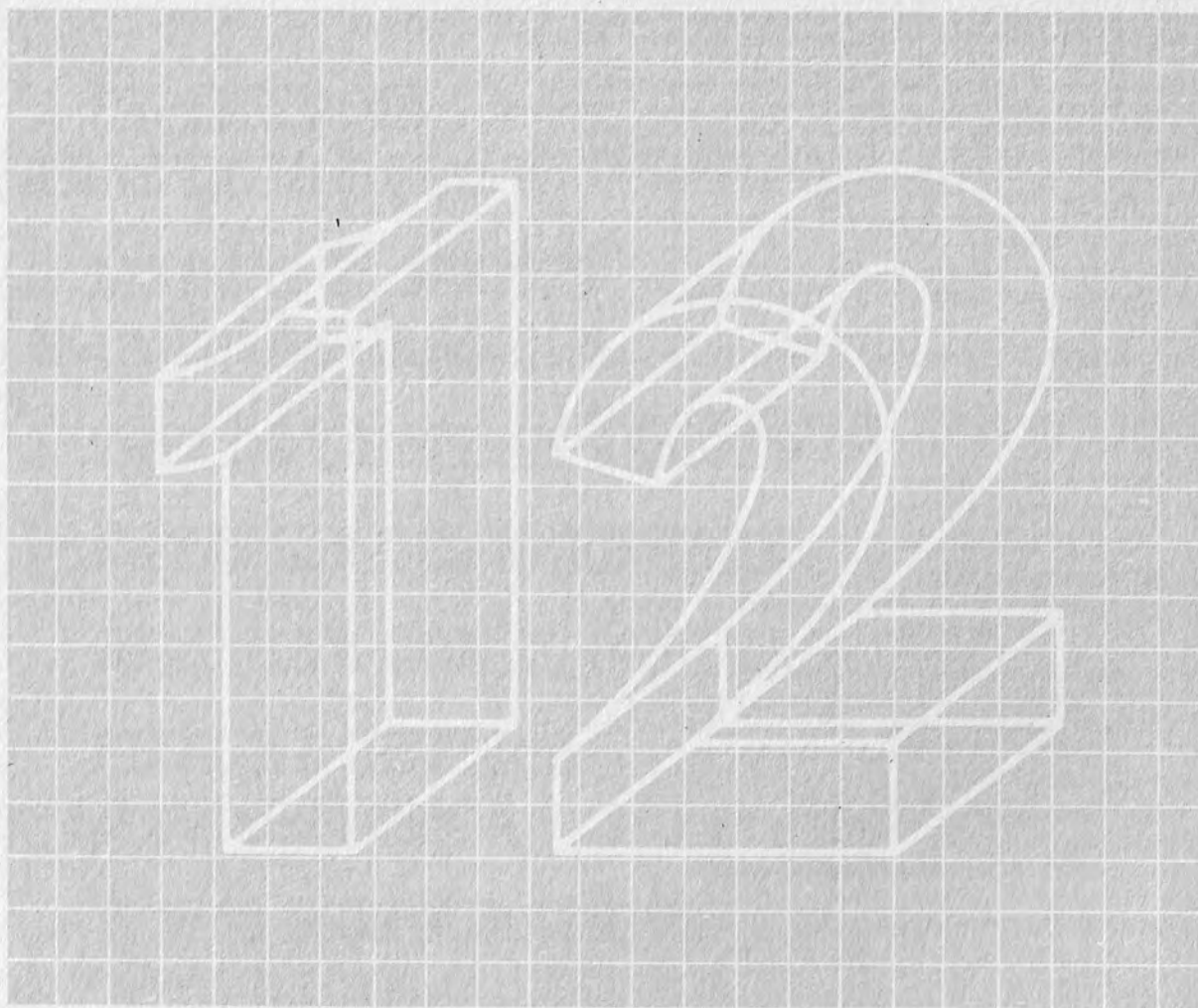
ATREVASE A HACER BRICOLAGE CON SU O.P.

EDICIONES SIGLO CULTURAL

BIBLIOTECA PRACTICA TALLER DE INFORMATICA

TRUCOS - SPRITES
BRICOLAGE DEL HARD
AULA ABIERTA
LOGO - TERMINOLOGIA

PROGRAMAS VALIDOS PARA AMSTRAD,
IBM, SPECTRUM, COMMODORE Y MSX



EDICIONES SIGLO CULTURAL

Una publicación de

EDICIONES SIGLO CULTURAL, S. A.

Director-editor:

RICARDO ESPAÑOL CRESPO.

Gerente:

ANTONIO G. CUERPO.

Directora de producción:

MARIA LUISA SUAREZ PEREZ.

Director de la colección:

JOSE ARTECHE, Ingeniero de Telecomunicación

Diseño:

BRAVO-LOFISH.

Maquetación:

D. S. M.

Dibujos:

JOSE OCHOA

Tomo 12 «Experiencias y prácticas en logo», Carlos Albert, Técnico de Informática; Adoración Llena Jubero, Técnico de Informática. «Manejo de sprites y elementos gráficos», «Trucos y rutinas básicas», Francisco Morales. Técnico de Informática. «Aprende con el ordenador», AULA DE INFORMATICA APLICADA: Soledad Tamariz-Martel, Diplomada en Telecomunicación; Francisco Blanca, Diplomado en Telecomunicación; Alejandro Marcos, Licenciado en Química; Fernando Suero, Diplomado en Telecomunicación. «El Taller del Hardware», Carlos Rey, Ingeniero Industrial. «Pequeña historia de la Informática», «Temas monográficos de vanguardia», «Terminología», «Vocabulario de Informática», Blanca Arbizu, Técnico de Informática.

Ediciones Siglo Cultural, S. A.

Dirección, redacción y administración:

Pedro Teixeira, 8-2.º. Teléf. 810 52 13. 28020 Madrid.

Publicidad:

Gofar Publicidad, S. A. Benito de Castro, 12 bis. 28028 Madrid.

Distribución en España:

COEDIS, S. A. Valencia, 245. Teléf. 215 70 97. 08007 Barcelona.

Delegación en Madrid: Serrano, 165. Teléf. 411 11 48.

Distribución en Ecuador: Muñoz Hnos.

Distribución en Perú: DISELPESA.

Distribución en Chile: Alfa Ltda.

Importador exclusivo Cono Sur:

CADE, S.R.L. Pasaje Sud América, 1532. Teléf.: 21 24 64.

Buenos Aires - 1.290. Argentina.

Todos los derechos reservados. Este libro no puede ser, en parte o totalmente, reproducido, memorizado en sistemas de archivo, o transmitido en cualquier forma o medio, electrónico, mecánico, fotocopia o cualquier otro, sin la previa autorización del editor.

ISBN del tomo: 84-7688-080-4

ISBN de la obra: 84-7688-047-2

Fotocomposición:

ARTECOMP, S. A. Albarracín, 50. 28037 Madrid.

Imprime:

MATEU CROMO. PINTO (Madrid).

© Ediciones Siglo Cultural, S. A., 1986

Depósito legal: M-43.185-1986

Printed in Spain - Impreso en España.

Suscripciones y números atrasados:

Ediciones Siglo Cultural, S. A.

Pedro Teixeira, 8-2.º. Teléf. 810 52 13. 28020 Madrid.

Abril, 1987.

P.V.P. Canarias: 355.—Ptas.

INDICE

■ EXPERIENCIA Y PRACTICAS EN LOGO	5
■ MANEJO DE SPRITES Y ELEMENTOS GRAFICOS	17
■ TRUCOS Y RUTINAS BASICAS	25
■ EL TALLER DEL HARDWARE	31
■ APRENDER CON EL ORDENADOR	39
■ PEQUEÑA HISTORIA DE LA INFORMATICA	51
■ TEMAS MONOGRAFICOS DE VANGUARDIA	53
■ VOCABULARIO DE INFORMATICA	59

EXPERIENCIAS Y PRACTICAS EN LOGO

■ Cómo definir procedimientos dentro del editor



A conocemos al editor del Logo. Esa pantalla oculta a la cual podemos acceder mediante la orden:

EDITA

En él podemos obtener el listado de las órdenes de un procedimiento especificándolo:

? EDITA "Nombre de procedimiento"

Vamos a ver cómo podemos definir los procedimientos directamente dentro del editor y qué otras cosas obtenemos en él, entrando de diferentes maneras.

Nada más comenzar a trabajar y antes de haber accedido alguna vez al editor, si introduces la orden:

?EDITA

o bien de su forma abreviada:

?ED

sin especificar un nombre de procedimiento, entramos en el editor Logo, pero en éste aparece únicamente el título:

EDITOR LOGO

y en la primera línea accesible, el cursor.

En esta situación podemos comenzar a definir nuestro procedimiento pero directamente en el editor. Bastaría con dar, en primer lugar, la orden PARA y a continuación especificar el nombre que queremos dar a nuestro procedimiento. Una vez realizado esto, podemos escribir las órdenes que lo van a componer. Primero nos situamos en la línea siguiente de donde nos encontramos, bien desplazando el cursor por medio de las teclas que existen para ello, o bien pulsando la tecla RETURN que lo situará directamente al comienzo de la línea siguiente.

A continuación comenzamos a teclear las órdenes. Cada vez que queramos colocarnos en una línea nueva, lo podemos realizar pulsando de nuevo RETURN. Cuando hayamos terminado, tenemos que poner como última orden FIN, igual que hacíamos cuando definíamos procedimientos fuera del editor.

Observa que en este caso ya no han aparecido ni la interrogación ni el signo >.

Una vez que consideremos que nuestro procedimiento ya contiene las órdenes necesarias, salimos del editor pulsando la tecla adecuada. Inmediatamente después, pasamos a la pantalla en la que estábamos antes de acceder al editor en el caso de los MSX, y al modo de pantalla texto en el caso de los PC-Compatibles y Spectrum.

En ambos casos recibimos el mensaje:

Nombre de procedimiento DEFINIDO

o bien

ACABAS DEFINIR Nombre de procedimiento

Los procedimientos con entrada utilizan la variable como dato.

EXPERIENCIAS Y PRACTICAS EN LOGO

Este mensaje nos indica que todo lo que hemos escrito en el editor ha sido almacenado en memoria formando un procedimiento. A partir de este momento podemos trabajar con el procedimiento definido de igual forma que hacíamos cuando lo definíamos fuera del editor.

Vamos a ver cómo podemos evitarnos el tener que escribir, cada vez que queramos ver en el editor un procedimiento determinado, su nombre.

Supongamos que definimos fuera del editor dos procedimientos:

```
?PARA A
>AV 20 GD 90
>REPITE 3 [AV 30 GI 120]
>FIN
```

```
?PARA B
>AV 20 GD 90
>REPITE 4 [AV 40 GD 90]
>FIN
```

Si ahora editamos uno de ellos, por ejemplo el A:

ED "A

entramos en el editor y en él aparecerá:

```
PARA A
AV 20 GD 90
REPITE 3 [AV 30 GI 120]
FIN
```

A continuación salimos del editor pulsando la tecla apropiada. Si ahora introducimos:

?ED

sin especificar el nombre de un procedimiento, volveríamos a entrar en el editor y en él aparecería otra vez el procedimiento A.

Es decir, si sólo damos la orden de editar sin especificar nombre de procedimiento alguno, entramos en el editor y en él aparece lo último que haya contenido.

Si quisiéramos editar el procedimiento B, sería necesario especificarlo:

?ED "B

y a partir de este momento cada vez que se introduzca simplemente la orden ED, obtendríamos el procedimiento B.

Veamos ahora qué ocurre si accedemos al editor de otra forma:

?ED "

Al no especificar un nombre de procedimiento pero sí las comillas, en el editor obtendríamos:

```
PARA
FIN
```

De esta forma podemos definir un nuevo procedimiento. Escribiríamos a continuación de PARA el nombre y comenzaríamos a teclear las órdenes. El FIN podemos, o bien desplazarlo hacia abajo situando el cursor sobre él y pulsando la tecla RETURN, o bien borrarlo, sin olvidarnos de volver a escribirlo al finalizar. Una vez que salimos del editor, el procedimiento queda definido.

Por último, podemos entrar en el editor de la siguiente forma:

pero en este caso, el nombre que se especifica no existe, es decir, no se ha definido ningún procedimiento con ese nombre.

Supongamos que ordenamos la edición del procedimiento C, el cual no ha sido anteriormente definido:

?ED "C

en el editor aparecería:

```
PARA C
FIN
```

Automáticamente obtenemos el procedimiento C, que al no estar definido aparece vacío. Al igual que en los casos anteriores, podemos proceder a definirlo aquí.

El Logo nos permite también poder obtener en el editor más de un procedimiento a la vez. Para ello basta con especificar los nombres de los procedimientos que queremos obtener de la siguiente forma:

?ED [Nombre procedimiento 1 Nombre procedimiento 2...]

Ahorra trabajo y memoria utilizando procedimientos con dos entradas.

Todos los procedimientos que se especifican aparecen en el editor en el mismo orden que se ha indicado y con una separación de una línea.

Esto puede ser de gran utilidad a la hora de estar trabajando con procedimientos que forman parte de un mismo proceso.

Al igual que podemos editar más de un procedimiento a la vez, podemos definir más de un procedimiento dentro del editor. Para ello lo único que hay que hacer es escribir un PARA y un FIN para cada uno de ellos.

El siguiente procedimiento nos permite dibujar en la pantalla cubos con las dimensiones que queramos. Para ello utilizamos la variable L, que es la que va a determinar la longitud de las aristas. En primer lugar, centramos el cubo, situando a la Tortuga a una distancia del centro, tanto en x como en y, igual a la mitad del lado:

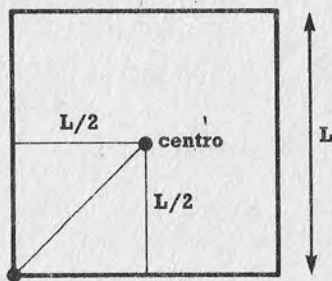


Fig. 1.

Estas son las órdenes que lo realizan:

```
?PARA CUBO :L
>SL
>BP
>PONX -:L / 2
>PONY -:L / 2
>BL
>REPITE 4 [AV :L GD 90]
>AV :L
>GD 45
>AV :L - (:L / 3)
>GD 45
>REPITE 4 [AV :L GD 90]
>AV :L
>GD 90
>AV :L
>GD 45
>AV :L - (:L / 3)
```

```
>GD 135
>AV :L
>GD 45
>AV :L - (:L / 3)
>SL
>PONX -:L / 2
>PONY -:L / 2
>BL
>AV :L - (:L / 3)
>FIN
```

Observa que la profundidad del cubo no es L, sino $L - 1/3 L$, para conseguir así dibujar un cubo proporcionado.
Si ahora lo ejecutamos:

?CUBO 40

Obtenemos:

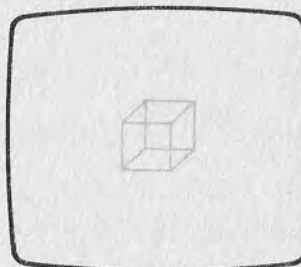


Fig. 2.

Si cambias la segunda orden del procedimiento (BP) por la orden CENTRO y a continuación lo ejecutamos dando los siguientes valores a la variable:

```
? CUBO 20
? CUBO 40
? CUBO 60
? CUBO 80
? CUBO 100
```

Obtenemos el siguiente dibujo:

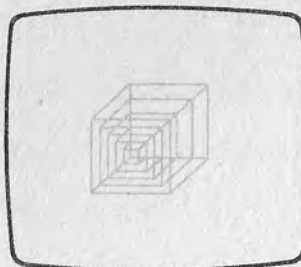


Fig. 3.

El denominador de una división no puede ser nunca 0.

EXPERIENCIAS Y PRACTICAS EN LOGO

Ahora introducimos una segunda variable al procedimiento para poder dibujar los cubos con la inclinación que queramos:

```
?PA RA CUBO :L :G
>SL
>BP
>PONX -:L / 2
>PONY -:L / 2
>BL
>GD :G
>REPITE 4 [AV :L GD 90]
>AV :L
>GD 45
>AV :L - ( :L / 3)
>GD 45
>REPITE 4 [AV :L GD 90]
>AV :L
>GD 90
>AV :L
>GD 45
>AV :L - ( :L / 3)
>GD 135
>AV :L
>GD 45
>AV :L - ( :L / 3)
>SL
>PONX -:L / 2
>PONY -:L / 2
>BL
>AV :L - ( :L / 3)
>FIN
```

Si ejecutamos el procedimiento dando los siguientes valores a las variables:

```
?CUBO 40 45
```

Obtenemos el cubo dibujado de la siguiente forma:

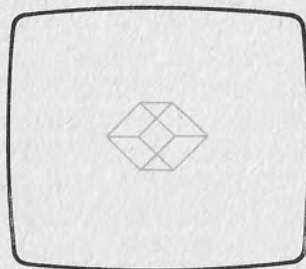


Fig. 4.

Por último, hacemos variable la profundidad del cubo:

```
?PARA CUBO :L :P
>SL
>BP
>PONX -:L / 2
>PONY -:L / 2
>BL
>REPITE 4 [AV :L GD 90]
>AV :L
>GD 45
>AV :L - ( :L / :P)
>GD 45
>REPITE 4 [AV :L GD 90]
>AV :L
>GD 90
>AV :L
>GD 45
>AV :L - ( :L / :P)
>GD 135
>AV :L
>GD 45
>AV :L - ( :L / :P)
>SL
>PONX -:L / 2
>PONY -:L / 2
>BL
>AV :L - ( :L / :P)
>FIN
```

Al ejecutarlo obtendríamos:

```
?CUBO 40 1.3
```

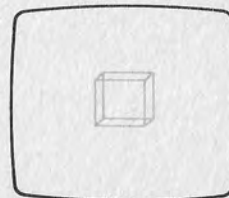


Fig. 5.

```
?CUBO 90 1.5
```

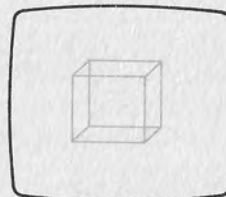


Fig. 6.

Podemos definir los procedimientos directamente en el editor.

■ Funciones matemáticas

A continuación exponemos una serie de funciones matemáticas generales, aunque no todas las versiones de LOGO dispongan de todas ellas.

-ENTERO n

Devuelve la parte entera de un número n truncando la parte decimal, es decir, sin redondear el número.

```
?ESCRIBE ENTERO 25.1
25
?ESCRIBE ENTERO 25.9
25
?ESCRIBE ENTERO -3.6
```

3

-REDONDEA n

Devuelve el número n redondeado al entero más próximo.

```
?ESCRIBE REDONDEA 5.4
5
?ESCRIBE REDONDEA 5.6
6
?ESCRIBE REDONDEA -3.2
-3
```

-ABS n

Devuelve el valor absoluto de un número.

```
?ESCRIBE ABS -30
30
?ESCRIBE ABS 30
30
?PONCL ABS CL -1
```

?AVANZA ABS COORX-COORY

La Tortuga avanzará la diferencia en valor absoluto de los valores que posean COORX y COORY.

-RESTO a b

Devuelve el resto de la división de a por b .

```
?ESCRIBE RESTO 20 10
0
?ESCRIBE RESTO 57 8
1
?ESCRIBE RESTO 40 0
IMPOSIBLE DIVIDIR POR CERO
```

-POTENCIA a b

Devuelve el número a elevado al número b .

Se deben tener en cuenta las reglas matemáticas:

- Si a es cero, b debe ser un número positivo.
- Si a es negativo, b debe ser un número entero.

```
?ESCRIBE POTENCIA 0 5
0
?ESCRIBE POTENCIA -5 -1
-0.2
?ESCRIBE POTENCIA -5 -1.5
POTENCIA NO ACEPTA [-5 -1.5]
COMO DATO
?ESCRIBE POTENCIA -4 7
-16384
?ESCRIBE POTENCIA 0 -3
POTENCIA NO ACEPTA [0 -3] COMO
DATO
```

Teclea este pequeño ejemplo en el que utilizamos la función POTENCIA.

```
?BP
?SL
?PONY 50
?BL
?REPITE 20 [PONCL CL +1 AV POTEN-
CIA 3 CL GD 90]
```

-RAIZCUADRADA n o RC N

Devuelve la raíz cuadrada del número

```
?ESCRIBE RAIZCUADRADA 25
5
?ESCRIBE RC 100
10
```

Si no se especifica un nombre de procedimiento cuando se introduce la orden EDITA, en el editor aparece lo último que haya contenido.

EXPERIENCIAS Y PRACTICAS EN LOGO

Si n es negativo, el LOGO muestra el mensaje:

RAIZCUADRADA NO ACEPTA n
COMO DATO

-EXP n

Función exponencial.

Devuelve el número e elevado a la potencia n . El número e es una constante matemática.

?ESCRIBE EXP -1

0.3678794

?ESCRIBE EXP 10

22026.7

?ESCRIBE EXP 0

1

?ESCRIBE EXP 1

2.718282

-LN n

El logaritmo neperiano es la función inversa de la exponencial.

Devuelve el logaritmo en base e de n . El número n debe ser positivo; en caso de que sea igual o menor que cero, el LOGO mostrará el mensaje:

LOGO NO ACEPTA n COMO DATO

?ESCRIBE LN 1

0

?ESCRIBE LN -3

LN NO ACEPTA -3 COMO DATO

?ESCRIBE LN 0

LN NO ACEPTA 0 COMO DATO

-PI

Devuelve la constante matemática PI.

La constante PI se utiliza para pasar los grados a radianes, y viceversa.

PI equivale a 180 grados.

2*PI equivale a 360 grados

?ESCRIBE PI

3.141592

?ESCRIBE (20*PI)/180

0.3490659

Esta última operación nos devuelve los radianes que corresponden a un ángulo de 20 grados.

Y ahora algunos ejemplos con procedimientos de dos entradas.

— *Semáforo de discos*

Primero definimos un procedimiento para dibujar el cuerpo del semáforo, con los discos sin colorear.

?PARA SEMA

>SL

>PONX -16

>BL PONCL 3

>REPITE 2 [AV 60 GD 90 AV 32 GD 90]

>GD 90 AV 19

>REPITE 2 [GD 90 AV 80 GD 90 AV 6]

>SL PONPOS [12 17]

>PONRUMBO 180

>BL

>REPITE 36 [AV 2 GD 10]

>SL PONPOS [12 44]

>BL

>REPITE 36 [AV 2 GD 10]

>FIN

Luego definimos un procedimiento para colorear los discos, los colores vienen dados por las dos variables de entrada de este procedimiento.

:A para el color del disco inferior

:B para el color del disco superior

?PARA COLOR :A :B

>SL

>PONPOS [5 16]

>BL

>PONCL :A

>RELLENA

>SL

>PONY 43

>BL

>PONCL :B

>RELLENA

>FIN

Y, por último, definimos un procedimiento para inicializar la pantalla y para llamar a los dos anteriores. Este procedimiento también es de dos entradas, puesto que hay una llamada al procedimiento COLOR.

?PARA DIBUJO :A :B

>BP

El dato de una función matemática puede ser una expresión aritmética.

```

>PONPALETA 0
>SEMA
>COLOR :A :B
>FIN

```

Si tu versión de LOGO no admite la orden PONPALETA, no la teclees.

Ejecutamos el dibujo por ejemplo:

```
?DIBUJO 1 2
```

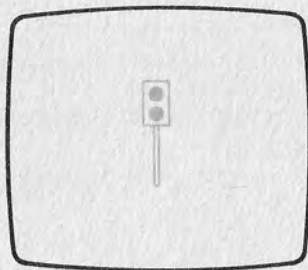


Fig. 7.

o también

```
?DIBUJO 2 3
```

— Colorear rectángulos

Definimos un procedimiento que enmarca diez rectángulos.

```

?PARA MARCO
>SL
>PONPOS [-125 -80]
>BL
>REPITE 2 [AV 160 GD 90 AV 250 GD 90]
>SL
>CENTRO
>FIN

```

El procedimiento RECTAN para dibujar los diez rectángulos.

```

?PARA RECTAN
>SL PONX -100
>BL
>REPITE 10 [REPITE 2 [AV 40 GD 90 AV 20 GD 90] GD 90 AV 20 GI 90]
>FIN

```

El procedimiento COLOR con dos entradas para seleccionar, dependiendo del va-

lor de :R, el rectángulo a pintar del color determinado por :C.

:R cuadrado a pintar de 1 a 10
:C color con el que rellenamos el rectángulo

```

?PARA COLOR :R :C
>SL
>PONY 10
>PONX -110 + :R * 20
>BL
>PONCL :C
>RELLENA
>FIN

```

Con DIBUJO inicializamos la pantalla y unimos todos los procedimientos definidos anteriormente.

```

?PARA DIBUJO :R :C
>BP
>PM
>PONCL 2
>MARCO
>PONCL 1
>RECTAN
>COLOR :R :C
>FIN

```

Para ejecutar el dibujo llamamos al procedimiento DIBUJO e indicamos dos valores.

```
?DIBUJO 3 2
```

Pintamos el tercer rectángulo del color que le corresponda al código 2.

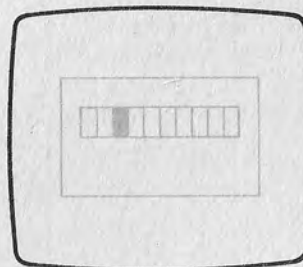


Fig. 8.

```
?DIBUJO 11 2
```

La función RC no admite como dato números negativos.

EXPERIENCIAS Y PRACTICAS EN LOGO

Rellenamos de color (2) el marco de los rectángulos.

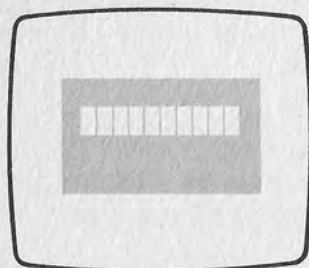


Fig. 9.

la del dibujo, teniendo en cuenta que podemos cambiar la altura y la anchura.

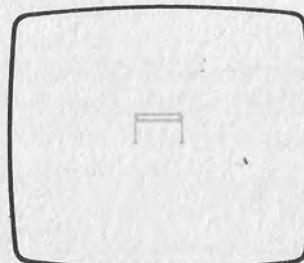


Fig. 10.

3. Intenta realizar el dibujo siguiente:

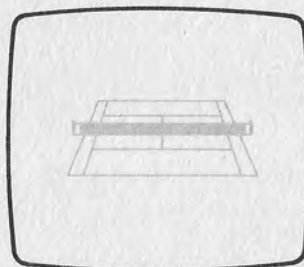


Fig. 11.

4. Define un procedimiento con dos entradas con el que poder obtener en la pantalla el siguiente dibujo en cualquier punto y con cualquiera inclinación.

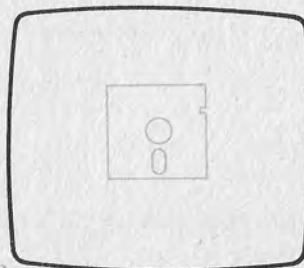


Fig. 12.

Resumen

- **ENTERO n**
Devuelve la parte entera de **n**.
- **REDONDEA n**
Devuelve el número **n** redondeado al entero más próximo.
- **ABS n**
Devuelve el valor absoluto de **n**.
- **RESTO a b**
Devuelve el resto de la división de **a** por **b**.
- **POTENCIA a b**
Devuelve el número **a** elevado a la potencia **b**.
- **RAIZCUADRADA n** o **RC n**
Devuelve la raíz cuadrada del número **n**.
- **EXP n**
Devuelve la constante matemática **e** elevada a **n**.
- **LN n**
Devuelve el logaritmo neperiano de **n**.
- **PI**
Devuelve el valor del número PI.

Ejercicios

1. Definir un procedimiento de dos entradas (:C y :H) que dibuje un triángulo rectángulo en el que uno de los catetos sea :C y la hipotenusa :H.

2. Define un procedimiento de dos entradas que nos permita dibujar una mesa como

5. ¿Son correctas las órdenes siguientes?

- ? ESCRIBE SUMA RC 16 20
- ? ESCRIBE RESTO 5 2 + 7
- ? ESCRIBE REDONDEA 20.7 + 90
- ? EDITA CASA
- ? ED "CASA:P
- ? ED "AV

El dato de la función LN debe ser mayor de 0.

Solución a los ejercicios

1:

Definición del procedimiento de dos entradas que dibuja un triángulo rectángulo.

:C uno de los catetos

:H la hipotenusa

? PARA TRIANG :C :H

> BP

> PM

> RE :C

> GD 90

> AV RC ((POTENCIA :H 2) - (POTENCIA :C 2))

> GI 135

> CENTRO

> FIN

El otro cateto lo calculamos hallando la raíz cuadrada de la diferencia entre la hipotenusa al cuadrado y el cateto al cuadrado.

Hipotenusa al cuadrado es igual a la suma de los cuadrados de los catetos.

? TRIAN 90 100

? TRIAN 40 50

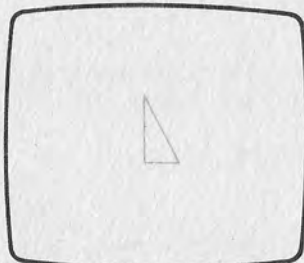


Fig. 13.

2:

Procedimiento de dos entradas para dibujar una mesa de diferentes tamaños.

:B para indicar la anchura de la mesa

:H para indicar la altura de la mesa

? PARA MESA :B :H

> BP

> PONCL 1

> SL

> PONPOS [-60 -60]

> PONRUMBO 90

> BL

> AV 2 RE 4 AV 2 GI 90

> AV :H RE :H

> SL

> PONX -60 + :B

> BL

> GD 90 AV 2 RE 4 AV 2 GI 90

> AV :H

> REPITE 2 [AV :H / 3 GI 90 AV :B GI 90]

> SL

> PONY COORY + 2

> PONX COORX - 2

> BL

> REPITE 2 [AV :H / 3 - 4 GI 90 AV :B - 4 GI 90]

> FIN

Lo ejecutamos por ejemplo:

? MESA 100 40

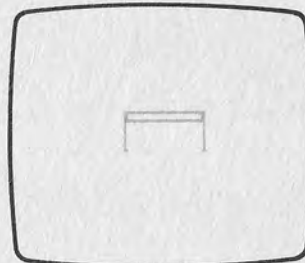


Fig. 14.

3:

Primero definimos un procedimiento para dibujar el contorno de la pista. La anchura de la pista la hemos fijado en 240 puntos en el plano principal y 160 en el plano de fondo.

? PARA PISTA

> BP

> SL

> PONPOS [-120 -50]

> BL

> PONPOS [-80 50]

> PONX 80

> PONPOS [120 -50]

> PONX -120

> SL

> PONPOS [-100 -50]

> BL

> PONPOS [-68 50]

> SL

> PONPOS [100 -50]

Podemos editar más de un procedimiento a la vez.

EXPERIENCIAS Y PRACTICAS EN LOGO

```
> BL
> PONPOS [68 50]
> SL
> PONPOS [-88 -15]
> BL
> PONX 88
> SL
> PONPOS [-73 30]
> BL
> PONX 73
> SL
> PONPOS [0 -15]
> BL
> PONY 30
> FIN
```

A continuación definimos el que dibuja la red apoyada sobre dos postes.

```
? PARA RED
> SL
> PONPOS [-110 5]
> BL
> REPITE 2 [AV 15 GD 90 AV 220 GD 90]
> RE 1
> REPITE 5 [AV 18 RE 18 GI 90 AV 1 GD 90] SL
> PONX 110
> BL
> REPITE 5 [AV 18 RE 18 GD 90 AV 1 GI 90] SL
> PONPOS [110 5]
> GI 45
> REPITE 42 [BL AV 20 SL RE 20 GI 45 AV 5 GD 45]
> SL
> PONPOS [-95 20]
> GI 90
> REPITE 42 [BL AV 20 SL RE 20 GI 135 AV 5 GD 135]
> FIN
```

Para ejecutarlo:

```
? PISTA RED
```

4:

Definimos un procedimiento que utiliza dos variables; P, que va a determinar el punto en donde queremos comenzar a dibujar el disquete y G, que determina la inclinación que

va a tener. El punto de comienzo va a tener el mismo valor en coordenada x e y.

```
? PARA DISQUETE :P :G
> BP
> SL
> PONX :P
> PONY :P
> GD :G
> BL
> REPITE 2 [AV 80 GD 90]
> AV 20 GD 90
> AV 5 GI 90
> AV 5 GI 90
> AV 5 GD 90
> AV 55 GD 90
> AV 80 GD 90
> SL
> AV 40 GD 90
> AV 30 GI 90
> BL
> REPITE 36 [AV 2 GD 10]
> SL
> RE 33 GD 90
> AV 5 GI 90
> BL
> REPITE 2 [AV 13 REPITE 9 [AV 2 GD 20] ]
> FIN
```

Si ahora lo ejecutamos con los siguientes valores para las variables:

```
? DISQUETE -40 0
```

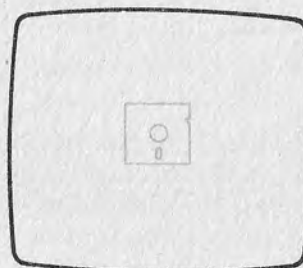


Fig. 15.

El disquete aparece en el centro de la pantalla y sin ninguna inclinación. Si ahora damos los valores:

```
? DISQUETE 30 45
```

Podemos editar un procedimiento que no esté definido pero éste aparecerá en blanco.

Obtendríamos el siguiente dibujo:

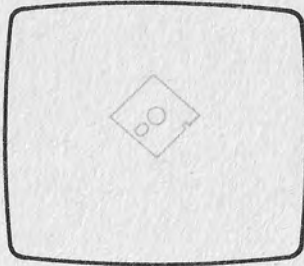


Fig. 16.

5:

? ESCRIBE SUMA RC 16 20

CORRECTO. El resultado es 24.

? ESCRIBE RESTO 5 2 + 7

CORRECTO. El resultado es 5, corresponde al resto de la división de 5 por 2+7.

? ESCRIBE REDONDEA SUMA 20.7 + 90

CORRECTO. Redondeamos la suma de 20.7 y 90, el resultado es 111.

? EDITA CASA

INCORRECTO. Falta indicar las comillas delante del nombre del procedimiento.

? ED "CASA:P

CORRECTO. Se editaría el procedimiento que recibe el nombre de CASA:P. Al no dejar espacio entre el nombre del procedimiento y la variable, identifica el nombre como CASA:P y no identifica la variable.

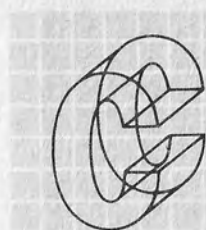
? ED "AV

INCORRECTO. Se está utilizando el nombre de una primitiva.

La función EXP es la inversa de la función LN.

En el editor se pueden definir a la vez más de un procedimiento.

MANEJO DE SPRITES Y ELEMENTOS GRAFICOS



OMO dijimos en el tomo anterior, vamos a ver un programa completo con movimiento, con control de este movimiento y con muchas cosas más que no hemos estudiado hasta ahora y que aprovecharemos para ver.

El juego en cuestión es el que aparece como programa 1 y le hemos llamado "El gusano loco".

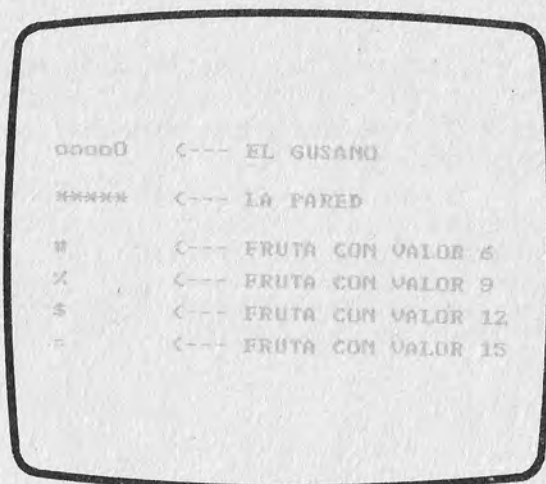


Fig. 1. Estas son las figuras que tienen los personajes de este programa. Intenta mejorarlas con ayuda de los caracteres definibles y de los SPRITES.

El juego consiste en comerse el máximo número de frutas posibles sin chocar con las paredes ni con nosotros mismos. Según vayamos comiendo fruta nuestro tamaño irá aumentando y el juego será más difícil cada vez. Cuando nos hayamos comido 15 frutas pasaremos a un nivel superior del juego.

Al principio del programa se te preguntará qué nivel quieres. Tienes que responder con un número entre el cero y el cuatro. La

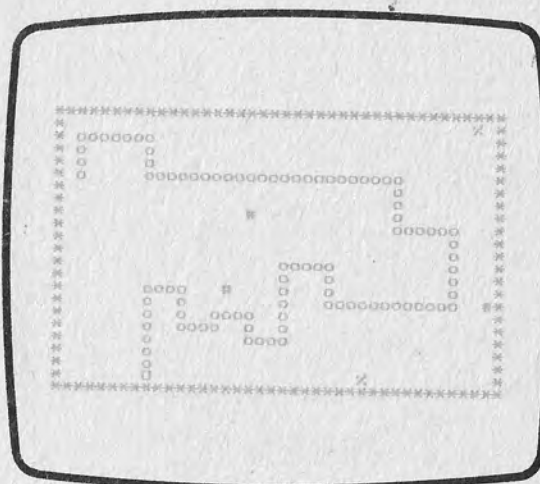


Fig. 2. Pantalla 1.

elección del nivel afecta a la velocidad del gusano al moverse por la pantalla. Si elegimos un nivel cero, el gusano se moverá más de prisa que en el nivel uno. Se recomienda empezar en el nivel tres o cuatro para coger práctica antes de jugar con el nivel cero o uno.

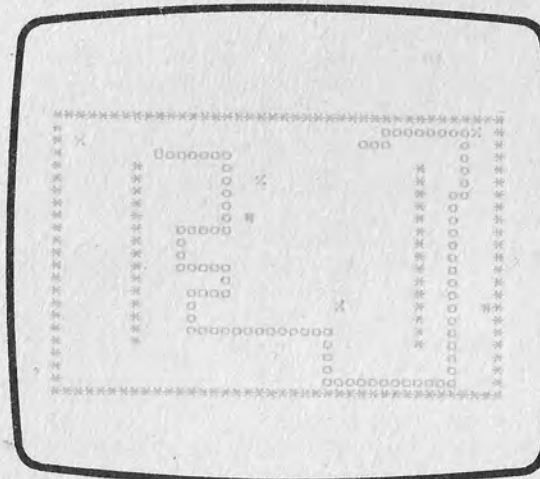


Fig. 3. Pantalla 2.

MANEJO DE SPRITES Y ELEMENTOS GRAFICOS

Internamente se han diseñado cuatro pantallas distintas. Se pasa de una a otra cuando, dentro de una pantalla, nos hemos comido 15 frutas. Cuando el jugador ha completado la cuarta fase del juego, el programa termina. Esto último se podría cambiar para que el juego continuase por la pantalla 1, pero a mayor velocidad. También se pueden diseñar más pantallas e incorporarlas al programa. Nosotros no lo hemos hecho para no hacer el programa excesivamente largo.

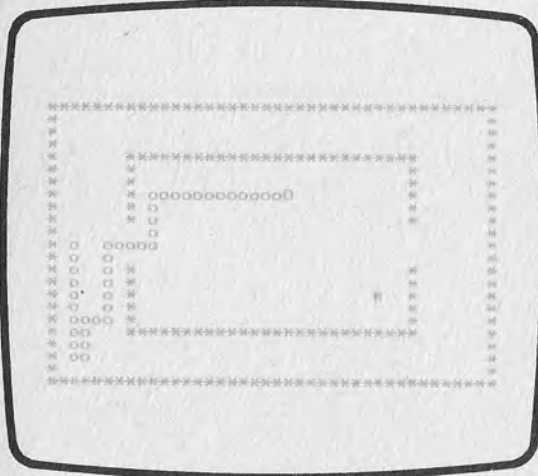


Fig. 4. Pantalla 3.

Se ha incluido una tabla de records que aparece cuando se han perdido las tres vidas de las que dispone el jugador o cuando han terminado las cuatro pantallas.

Aparte de la dificultad que supone el

juego en sí, hay otra que lo hace más interesante. Cuando el gusano es muy largo puede que se fragmente debido a la muda que todos los años sufren dichos animales. En tal caso, parte del cuerpo del gusano permanecerá en pantalla como parte muerta, pero de la que has de tener mucho cuidado, pues si te chocas con ella, perderás una vida. También se pierde una vida al chocar contra las paredes.

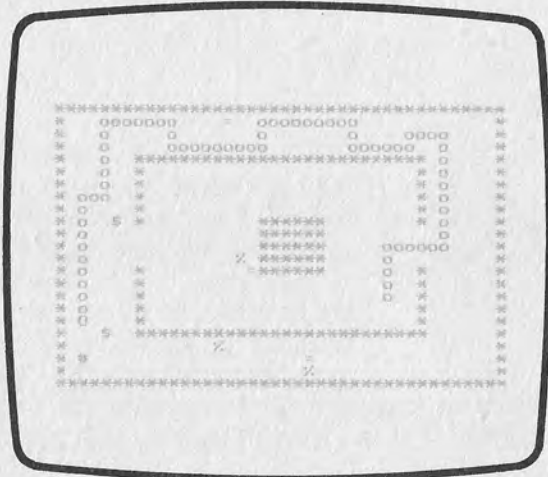


Fig. 5. Pantalla 4.

Por lo demás, nada más. A continuación aparece el programa y las modificaciones que hay que hacerle para que funcione en ordenadores distintos del IBM.

Se recomienda a los usuarios del IBM y del AMSTRAD que ejecuten este programa en el modo de 40 columnas por línea.

```

100 REM *****
110 REM * JUEGO DEL GUSANO LOCO *
120 REM *****
130 REM
140 REM *****
150 REM * PANTALLA DE PRESENTACION *
160 REM *****
170 REM
180 CLS
190 PRINT "GUSANO LOCO"
200 PRINT "-----"
210 PRINT:PRINT:PRINT
220 PRINT "USA LAS SIGUIENTES TECLAS:"
230 PRINT:PRINT
240 PRINT "O - IZQUIERDA"
250 PRINT "P - DERECHA"
260 PRINT "Q - ARRIBA"
270 PRINT "A - ABAJO"
280 PRINT:PRINT
290 INPUT "QUE NIVEL QUIERES (0-4) ";NN
300 IF NN<0 OR NN>4 THEN GOTO 290

```

```

310 LET NN=25*NN
320 REM
330 REM *****
340 REM * INICIALIZACION DEL PROGRAMA *
350 REM *****
360 REM
370 CLS
380 PRINT "ESPERA UN MOMENTO"
390 DIM A$(10):DIM S(10):DIM F$(4)
400 FOR I=1 TO 10
410   LET A$(I)="S. Cultural"
420   LET S(I)=100
430 NEXT I
440 LET NV=1:LET VV=3
450 DIM A(40,23)
460 LET F$(1)="#"
470 LET F$(2)="#"
480 LET F$(3)="$"
490 LET F$(4)="-"
500 REM
510 REM *****
520 REM * PROGRAMA GENERAL *
530 REM *****
540 REM
550 FOR I=1 TO 40
560   FOR J=1 TO 23
570     LET A(I,J)=0
580   NEXT J
590 NEXT I
600 LO=0
610 CLS
620 LET X=10:LET Y=10
630 LET X1=10:LET Y1=10
640 LET X3=10:LET Y3=10
650 GOSUB 710
660 ON NV GOSUB 900,960,1100,1360
670 GOSUB 1720
680 GOSUB 2040
690 FOR J=1 TO NN:NEXT J
700 GOTO 670
710 REM
720 REM *****
730 REM * DIBUJO DE LOS LIMITES DE LA PANTALLA *
740 REM *****
750 REM
760 FOR I=1 TO 40
770   PRINT "*";
780   LET A(I,1)=1
790 NEXT I
800 FOR I=2 TO 22
810   PRINT "*";TAB(40);"*"
820   LET A(1,I)=1
830   LET A(40,I)=1
840 NEXT I
850 FOR I=1 TO 40
860   PRINT "*";
870   LET A(I,23)=1
880 NEXT I
890 RETURN
900 REM
910 REM *****
920 REM * PANTALLA No. 1 *
930 REM *****
940 REM

```

MANEJO DE SPRITES Y ELEMENTOS GRAFICOS

```
950 RETURN
960 REM
970 REM *****
980 REM * PANTALLA No. 2 *
990 REM *****
1000 REM
1010 FOR I=5 TO 19
1020   LOCATE I,8
1030   PRINT "*"
1040   LET A(8,I)=1
1050   LOCATE I,33
1060   PRINT "*"
1070   LET A(33,I)=1
1080 NEXT I
1090 RETURN
1100 REM
1110 REM *****
1120 REM * PANTALLA No. 3 *
1130 REM *****
1140 REM
1150 FOR I=8 TO 33
1160   LOCATE 5,I
1170   PRINT "*"
1180   LET A(I,5)=1
1190   LOCATE 19,I
1200   PRINT "*"
1210   LET A(I,19)=1
1220 NEXT I
1230 FOR I=6 TO 18
1240   LOCATE I,8
1250   PRINT " ";TAB(33);"*"
1260   LET A(8,I)=1
1270   LET A(33,I)=1
1280 NEXT I
1290 FOR I=11 TO 13
1300   LOCATE I,8
1310   PRINT " ";TAB(33);" "
1320   LET A(8,I)=0
1330   LET A(33,I)=0
1340 NEXT I
1350 RETURN
1360 REM
1370 REM *****
1380 REM * PANTALLA No. 4 *
1390 REM *****
1400 REM
1410 GOSUB 1100
1420 FOR I=10 TO 14
1430   LOCATE I,19
1440   PRINT "*****"
1450   FOR J=19 TO 24
1460     LET A(J,I)=1
1470   NEXT J
1480 NEXT I
1490 RETURN
1500 REM
1510 REM *****
1520 REM * CHOQUE CONTRA LA PARED *
1530 REM *****
1540 REM
1550 CLS
1560 PRINT "TE HAS CHOCADO CONTRA LA PARED."
1570 PRINT "LA PROXIMA VEZ HAS DE TENER MAS CUIDADO"
1580 IF VV=1 THEN GOTO 2450
```

```

1590 LET VV=VV-1
1600 GOTO 550
1610 REM
1620 REM *****
1630 REM * CHOQUE CONTRA UNO MISMO *
1640 REM *****
1650 REM
1660 CLS
1670 PRINT "TE HAS CHOCADO CONTIGO MISMO"
1680 PRINT "QUE TE PASA. (ESTAS TONTO?"
1690 IF VV=1 THEN GOTO 2450
1700 LET VV=VV-1
1710 GOTO 550
1720 REM
1730 REM *****
1740 REM * COLOCACION DE FRUTAS *
1750 REM *****
1760 REM
1770 IF RND>.1 THEN RETURN
1780 LET V=INT(RND*4+1)
1790 LET X2=INT(RND*40+1)
1800 LET Y2=INT(RND*23+1)
1810 IF A(X2,Y2)<>0 THEN RETURN
1820 LET A(X2,Y2)=V+1
1830 LOCATE Y2,X2:PRINT F$(V)
1840 RETURN
1850 REM
1860 REM *****
1870 REM * COMPROBACION DE CHOQUE *
1880 REM *****
1890 REM
1900 IF A(X,Y)=0 THEN RETURN
1910 IF A(X,Y)=-1 THEN GOTO 1610
1920 IF A(X,Y)=1 THEN GOTO 1500
1930 LET SC=SC+15*A(X,Y):LO=A(X,Y)*3
1940 IF NC<>15 THEN LET NC=NC+1:RETURN
1950 CLS:LET NC=0
1960 PRINT "MUY BIEN. HAS TERMINADO CON LA FASE ";NV
1970 IF NV=4 THEN GOTO 2010
1980 PRINT "PREPARATE PARA LA SIGUIENTE"
1990 LET NV=NV+1
2000 GOTO 550
2010 PRINT "TAMBIEN HAS TERMINADO CON EL JUEGO"
2020 FOR I=1 TO 3000:NEXT I
2030 GOTO 2450
2040 REM
2050 REM *****
2060 REM * MOVIMIENTO DEL GUSANO *
2070 REM *****
2080 REM
2090 LET B$=INKEY$
2100 IF B$<>"P" AND B$<>"O" AND B$<>"Q" AND B$<>"A" THEN LET B$=C$
2110 LET C$=B$:LET X1=X:LET Y1=Y
2120 IF B$="P" THEN LET X=X+1
2130 IF B$="O" THEN LET X=X-1
2140 IF B$="Q" THEN LET Y=Y-1
2150 IF B$="A" THEN LET Y=Y+1
2160 GOSUB 1860
2170 LOCATE Y1,X1
2180 PRINT "o"
2190 LOCATE Y,X
2200 PRINT "O"
2210 LET A(X,Y)=-1
2220 IF LO=0 THEN GOTO 2250

```

MANEJO DE SPRITES Y ELEMENTOS GRAFICOS

```

2230 LET LO=LO-1
2240 RETURN
2250 LOCATE Y3,X3
2260 PRINT " "
2270 LET A(X3,Y3)=0
2280 IF A(X3+1,Y3)=-1 THEN LET X3=X3+1:RETURN
2290 IF A(X3-1,Y3)=-1 THEN LET X3=X3-1:RETURN
2300 IF A(X3,Y3-1)=-1 THEN LET Y3=Y3-1:RETURN
2310 IF A(X3,Y3+1)=-1 THEN LET Y3=Y3+1:RETURN
2320 RETURN
2330 REM
2340 REM *****
2350 REM * (OTRA PARTIDA? *
2360 REM *****
2370 REM
2380 PRINT:PRINT
2390 PRINT "(OTRA PARTIDA? (S/N) "
2400 LET A$=INKEY$:IF A$="" THEN GOTO 2400
2410 IF A$="S" OR A$="s" THEN LET NV=1:LET VV=3:LET SC=0:GOTO 550
2420 IF A$="N" OR A$="n" THEN GOTO 2960
2430 GOTO 2400
2440 LET LO=LO-1
2450 REM
2460 REM *****
2470 REM * TABLA DE GANADORES *
2480 REM *****
2490 REM
2500 FOR I=1 TO 1000
2510 NEXT I
2520 CLS:PRINT "PUNTUACIONES DE HOY"
2530 PRINT:PRINT "-----"
2540 FOR I=4 TO 17
2550   LOCATE I,1
2560   PRINT "!";TAB(40);"! "
2570 NEXT I
2580 PRINT "-----"
2590 IF SC>=S(10) THEN LET S(10)=SC:LET A$(10)=CHR$(254)
2600 FOR I=1 TO 10
2610   FOR J=1 TO I
2620     IF S(J)<S(I) THEN LET A=S(J):LET S(J)=S(I):LET S(I)=A:LET B$=A$(J):LE
T A$(J)=A$(I):LET A$(I)=B$
2630   NEXT J
2640 NEXT I
2650 FOR I=1 TO 10
2660   IF A$(I)=CHR$(254) THEN LET PP=I
2670 NEXT I
2680 IF PP=0 THEN GOTO 2700
2690 A$(PP)="....."
2700 FOR I=1 TO 10
2710   LOCATE 5+I,3
2720   PRINT A$(I);TAB(30);S(I)
2730 NEXT I
2740 LOCATE 5+PP,3,1
2750 LET A$(PP)=""
2760 IF PP=0 THEN GOTO 2850
2770 FOR I=1 TO 15
2780   LET A$=INKEY$:IF A$="" THEN GOTO 2780
2790 IF A$=CHR$(8) AND I>1 THEN PRINT CHR$(29);".";CHR$(29);:LET I=I-1:LET A$(PP
)=MID$(A$(PP),1,I)
2800 IF A$=" " THEN GOTO 2830
2810 IF A$=CHR$(13) THEN LET I=15:GOTO 2840
2820 IF A$<"0" OR A$>"z" THEN GOTO 2780
2830 PRINT A$;:LET A$(PP)=A$(PP)+A$
2840 NEXT I

```

```

2850 LOCATE 1,1,0
2860 LOCATE 20,1
2870 PRINT "<<< PULSA UNA TECLA >>>"
2880 LET A$=INKEY$:IF A$="" THEN GOTO 2880
2890 LOCATE 20,1
2900 GOTO 2330
2910 REM
2920 REM *****
2930 REM * FIN DEL JUEGO *
2940 REM *****
2950 REM
2960 CLS
2970 PRINT "HASTA PRONTO"
2980 PRINT "=====
2990 FOR I=1 TO 10
3000 PRINT
3010 NEXT I
3020 END

```

Cultural	Puntuación
Cultural	100
Cultural	90
Cultural	80
Cultural	70
Cultural	60
Cultural	50
Cultural	40
Cultural	30

Fig. 6. Tabla de las máximas puntuaciones.

COMMODORE:

```

180 PRINT CHR$(147)
370 PRINT CHR$(147)
610 PRINT CHR$(147)
1020 POKE 214,I-1:POKE 211,7
1050 POKE 214,I-1:POKE 211,32
1160 POKE 214,4:POKE 211,I-1
1190 POKE 214,18:POKE 211,I-1
1240 POKE 214,I-1:POKE 211,7
1300 POKE 214,I-1:POKE 211,7
1430 POKE 214,I-1:POKE 211,18
1550 PRINT CHR$(147)
1660 PRINT CHR$(147)
1770 IF RND (0)>.1 THEN RETURN
1780 LET V=INT(RND(0)*4+1)
1790 LET X2=INT(RND(0)*40+1)
1800 LET Y2=INT(RND(0)*23+1)
1830 POKE 214,Y2-1:POKE 211,
X2-1:PRINT F$(V)

```

SPECTRUM:

```

1950 PRINT CHR$(147):LET NC=0
2090 GET B$
2170 POKE 214,Y1-1:POKE 211,X1-1
2190 POKE 214,Y1-1:POKE 211,X-1
2250 POKE 214,Y3-1:POKE 211,X3-1
2400 GET A$:IF A$="" THEN GOTO 2400
2520 PRINT CHR$(147); "PUNTUACIONES DE HOY"
2550 POKE 214,I-1:POKE 211,0
2560 PRINT "!", TAB(39);"!"
2710 POKE 214,4+1:POKE 211,2
2740 POKE 214,4+PP:POKE 211,2
2780 GET A$:IF A$="" THEN GOTO 2780
2850 REM
2860 POKE 214,19:POKE 211,0
2880 GET A$:IF A$="" THEN GOTO 2880
2960 PRINT CHR$(147)

```

```

90 LET B$="":LET C$="":SC=0
390 DIM A$(10,15):DIM S(10):DIM
F$(4,1)
760 FOR I=1 TO 32
800 FOR I=2 TO 20
810 PRINT ":",TAB(31);"::"
850 FOR I=1 TO 32
870 LET A(I,21)=1
1010 FOR I=5 TO 17
1020 PRINT AT I-1,5;
1040 LET A(6,I)=1
1050 PRINT AT I-1,27;
1070 LET A(28,I)=1
1150 FOR I=6 TO 28
1160 PRINT AT 4,I-1;
1190 PRINT AT 16,I-1;
1210 LET A (I,17)=1

```

MANEJO DE SPRITES Y ELEMENTOS GRAFICOS

```
1230 FOR I=6 TO 16
1240 PRINT AT I-1,5;
1250 PRINT " ";TAB(26);" "
1260 LET A(6,I)=1
1270 LET A(27,I)=1
1300 PRINT AT I-1,5
1310 PRINT " "; TAB(26);" "
1320 LET A(6,I)=0
1330 LET A(27,I)=0
1420 FOR I=9 TO 13
1430 PRINT AT I-1,15
1450 FOR J=16 TO 21
1830 PRINT AT Y2-1,X2-1;F$(V)
1930 LET SC=SC+15*A(X,Y):LET
    LO=A(X,Y)*3
2170 PRINT AT Y1-1,X1-1;
2190 PRINT AT Y1-1,X-1;
2250 PRINT AT Y3-1,X3-1;
2550 PRINT AT I,0;
2560 PRINT " ";TAB(31);" "
2710 PRINT AT 4+1,2;
2720 PRINT A$(I);TAB(25);S(I)
2740 PRINT AT 4+PP,2;
2850 REM
2860 PRINT AT 20,0;
2890 PRINT AT 20,0;
```

AMSTRAD:

```
1020 LOCATE 8,I
1050 LOCATE 33,I
1160 LOCATE I,5
1190 LOCATE I,19
1240 LOCATE 8,I
1300 LOCATE 8,I
1430 LOCATE 19,I
1830 LOCATE X2,Y2
2170 LOCATE X1,Y1
2190 LOCATE X,Y
```

```
2250 LOCATE X3,Y3
2550 LOCATE I,I
2710 LOCATE 3,5+I
2740 LOCATE 3,5+PP,1
2860 LOCATE 1,20
2890 LOCATE 1,20
```

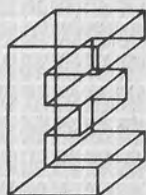
MSX:

```
1020 LOCATE 8,I
1050 LOCATE 33,I
1160 LOCATE I,5
1190 LOCATE I,19
1240 LOCATE 8,I
1300 LOCATE 8,I
1430 LOCATE 19,I
1770 IF RND(0)>.1 THEN RETURN
1780 LET V=INT(RND(0)*4+1)
1790 LET X2=INT(RND(0)*40+1)
1880 LET Y2=INT(RND(0)*23+1)
1830 LOCATE X2,Y2
2170 LOCATE X1,Y1
2190 LOCATE X,Y
2250 LOCATE X3,Y3
2550 LOCATE I,I
2710 LOCATE 3,5+I
2740 LOCATE 3,5+PP,1
2860 LOCATE 1,20
2890 LOCATE 1,20
```

Por la amplitud del programa se deja la explicación del mismo para el siguiente tomo.

Como el programa, como se puede apreciar, no es ninguna maravilla en lo que a gráficos se refiere, se pide al lector que intente modificar el programa para que, utilizando la capacidad de cada ordenador para definir caracteres, el gusano, la pared y las frutas sean, en realidad, lo que pretenden.

TRUCOS Y RUTINAS BASICAS



N el tomo anterior vimos el programa gestor de ficheros en su totalidad. Debido a la falta de espacio, no se dieron las modificaciones necesarias para que el programa funcionase en ordenadores distintos del IBM. A

continuación aparecen dichas modificaciones:

COMMODORE:

```

240 PRINT CHR$(147)
330 PRINT CHR$(147)
520 PRINT CHR$(147)
590 POKE 214,19:POKE 211,0
610 GET A$:IF A$="" THEN GOTO 610
750 PRINT CHR$(147)
950 PRINT CHR$(147)
1080 PRINT CHR$(147)
1300 PRINT CHR$(147)
1550 PRINT CHR$(147)
1640 OPEN 1,8,1,B$
1810 OPEN 1,1,1,B$
2090 PRINT CHR$(147)
2170 OPEN 1,8,0,B$
2360 OPEN 1,1,0,B$
2660 PRINT CHR$(147)
2740 POKE 214,4:POKE 211,0
2975 OPEN 1,4:CMD 1
3040 NEXT V:PRINT
      TAB(VAL(N$(CR,3))):LET
      CR=CR+1
3070 PRINT MID$(F$(Z),V,1);
3100 IF N$(1,4)="-1" THEN PRINT
      CHR$(147):GOSUB 3150 GOTO
      3120
3110 FOR R=1 TO VAL(N$(1,4)):
      PRINT:NEXT R

```

```

3130 "PRINT "-----"
3140 CLOSE #1:RETURN
3170 PRINT TAB(VAL(N$(S,3))):
      M$(VAL(N$(S,2)));
3180 NEXT S:PRINT
      "-----"
3230 PRINT CHR$(147)
3280 GET A$
3330 PRINT CHR$(147)
3440 PRINT CHR$(147)
3470 POKE 214,4:POKE 211,0
3490 POKE 214,11:POKE 211,0
3530 POKE 214,4:POKE 211,0
3550 POKE 214,8:POKE 211,0
3580 POKE 214,11:POKE 211,0
3590 PRINT "      ":REM 10 ESPACIOS
3600 POKE 214,11:POKE 211,0
3630 POKE 214,8:POKE 211,21
3680 PRINT "<CURSOR-IZQ>"," "
3690 POKE 214,4:POKE 211,0
3730 POKE 214,9:POKE 211,0
3760 POKE 214,11:POKE 211,0
3770 PRINT "      ":REM 10 ESPACIOS
3780 POKE 214,11:POKE 211,0
3810 POKE 214,9:POKE 211,21
3860 PRINT "<CURSOR-IZQ>"," "
3870 POKE 214,11:POKE 211,0
3960 PRINT CHR$(147)
4010 PRINT "<CURSOR-IZQ>"," "
4080 PRINT "<CURSOR-IZQ>"," "
4090 POKE 214,11:POKE 211,0
4150 POKE 214,14:POKE 211,0
4170 GET A$
4200 REM
4230 POKE 214,Z-1:POKE 211,0
4240 FOR Q=1 TO 40:PRINT " ";:NEXT Q
4260 POKE 214,4:POKE 211,0
4310 POKE 214,11:POKE 211,0
4370 POKE 214,Z-1:POKE 211,0

```

TRUCOS Y RUTINAS BASICAS

```

4380 FOR Q=1 TO 40:PRINT " ";:NEXT Q
4400 POKE 214,4:POKE 211,0
4440 POKE 214,19:POKE 211,0
4460 GET A$
4510 POKE 214,Z-1:POKE 211,0
4520 FOR Q=1 TO 40:PRINT " ";:NEXT Q
4540 POKE 214,4:POKE 211,0
4580 POKE 214,11:POKE 211,0
4610 PRINT CHR$(147)
4660 PRINT CHR$(147)
4740 POKE 214,Y:POKE 211,X
4780 POKE 214,Y:POKE 211,X
4800 POKE 214,Y:POKE 211,X
4810 GET A$
4830 IF A$=CHR$(20) AND L0>0 THEN
    LET L0=L0-1:LET
    D$=LEFT$(D$,L0):LET X=X-1:PO-
    KE 214,Y:POKE 211,X:PRINT
    "-":POKE 214,Y:POKE 211,X:GOTO
    4810
4890 POKE 214,Y:POKE 211,X
4930 POKE 214,Y:POKE 211,X
5050 POKE 214,Y:POKE 211,X
5070 POKE 214,Y:POKE 211,X+19
5080 PRINT CHR$(102);
5090 GET A$
5140 POKE 214,Y:POKE 211,X+19
5160 GET A$
5220 POKE 214,Y:POKE 211,X
5230 FOR Q=1 TO L+3:PRINT " ";:NEXT
    Q
5280 POKE 214,Y:POKE 211,X+19
5320 POKE 214,Y:POKE 211,X+19
5390 POKE 214,Y:POKE 211,X+19
5510 REM
5530 POKE 214,Y:POKE 211,X
5550 GET B$:IF B$<>" " THEN GOTO
    5650
5580 POKE 214,Y:POKE 211,X
5590 FOR Q=1 TO L:PRINT " ";:NEXT Q
5600 GET B$:IF B$<>" " THEN GOTO
    5650
5650 POKE 214,Y:POKE 211,X
5660 FOR Q=1 TO L:PRINT " ";:NEXT Q
5940 PRINT CHR$(147)
6010 PRINT CHR$(147)
6030 POKE 214,1:POKE 211,X
6050 POKE 214,2:POKE 211,X
6160 POKE 214,2*Z+3:POKE 211,X
6210 PRINT CHR$(147)
6280 PRINT CHR$(147)
6390 PRINT CHR$(147)
6410 PRINT CHR$(147)
6550 PRINT CHR$(147)
6670 PRINT CHR$(147)
6770 REM

```

```

6850 PRINT CHR$(147):LET Z=NF
6940 PRINT CHR$(147)
7030 PRINT CHR$(147)
7100 PRINT CHR$(147)
7180 PRINT CHR$(147);
    "O P E R A C I O N
    T E R M I N A D A"
7230 PRINT CHR$(147)
7300 PRINT CHR$(147)
7420 POKE 214,8:POKE 211,0
7480 POKE 214,15:POKE 211,0
7500 POKE 214,17:POKE 211,0
7520 POKE 214,15:POKE 211,0
7550 POKE 214,17:POKE 211,0
7590 PRINT CHR$(147)
7670 PRINT CHR$(147)
7780 PRINT CHR$(147)
7800 OPEN 1,1,0,N$
7900 CLOSE #1:PRINT CHR$(147)
8100 POKE 214,9:POKE 211,0
8170 POKE 214,19:POKE 211,0
8190 GET A$:IF A$="" THEN GOTO 8190
8230 POKE 214,9:POKE 211,0
8240 FOR Q=1 TO 80:PRINT " ";:NEXT Q
8250 POKE 214,19:POKE 211,0
8260 FOR Q=1 TO 80:PRINT " ";:NEXT Q
8340 PRINT CHR$(147):LET CC=1

```

SPECTRUM:

```

220 DIM F$(200,100):DIM
    A$(10,20):DIM N(20):DIM
    N$(20,4,3)
470 GOTO 490*(A=1)+650*(A=2)+
    1050*(A=3)+1150*(A=4)+
    145*(A=5)+1990*(A=6)+
    2550*(A=7)+3200*(A=8)
590 PRINT AT 19,0;
760 LET A=VAL(A$):GOTO
    770*(A=1)+850*(A=2)+
    890*(A=3)+980*(A=4)
1340 LET A=VAL(A$):GOTO
    1350*(A=1)+1390*(A=2)+
    1430*(A=3)
1600 IF LEN(B$)>10 THEN GOTO 1590
1640 SAVE "M",1;B$+"1" DATA N()
1650 SAVE "M",1;B$+"2" DATA M$()
1660 SAVE "M",1;B$+"3" DATA F$()
1670 SAVE "M",1;B$+"4" DATA N$()
1680 REM
1690 REM
1700 REM
1710 REM
1720 REM
1730 REM
1740 REM
1750 REM

```

```

1760 REM
1770 REM
1780 REM
1790 REM
1810 SAVE B$+"1" DATA N()
1820 SAVE B$+"2" DATA M$()
1830 SAVE B$+"3" DATA F$()
1840 SAVE B$+"4" DATA N$()
1850 REM
1860 REM
1870 REM
1880 REM
1890 REM
1900 REM
1910 REM
1920 REM
1930 REM
1940 REM
1950 REM
1960 REM
1970 REM
2140 IF LEN (B$)>10 THEN GOTO 2130
2170 LOAD "M";1;B$+"1" DATA N()
2180 LOAD "M";1;B$+"2" DATA M$()
2190 LOAD "M";1;B$+"3" DATA F$()
2200 LOAD "M";1;B$+"4" DATA N$()
2210 FOR Q=1 TO 200
2220 IF F$(Q)="" THEN LET TT=Q-1:LET
      Q=200
2230 NEXT Q
2240 REM
2250 REM
2260 REM
2270 REM
2280 REM
2290 REM
2300 REM
2310 REM
2320 REM
2330 REM
2340 REM
2360 LOAD B$+"1" DATA N()
2370 LOAD B$+"2" DATA M$()
2380 LOAD B$+"3" DATA F$()
2390 LOAD B$+"4" DATA N$()
2400 FOR Q=1 TO 200
2410 IF F$(Q)="" THEN LET TT=Q-1:LET
      Q=200
2420 NEXT Q
2430 REM
2440 REM
2450 REM
2460 REM
2470 REM
2480 REM
2490 REM

```

```

2500 REM
2510 REM
2520 REM
2530 REM
2740 PRINT AT 4,0;
3030 IF F$(Z,V)=CHR$(254) THEN LET
      CD=CD+1:IF CD=CC THEN LET
      Z1=V+1:LET V=LEN(F$(Z))
3060 IF F$(Z,V)=CHR$(254) THEN LET
      V=LEN(F$(Z)):GOTO 3080
3070 LPRINT F$(Z,V);
3470 PRINT AT 4,0;
3490 PRINT AT 11,0;
3500 INPUT LINE "=="> "A$
3530 PRINT AT 4,0;
3550 PRINT AT 8,0;
3580 PRINT AT 11,0;
3590 PRINT "      ":REM 10 ESPACIOS
3600 PRINT AT 11,0;
3610 INPUT LINE "=="> "N$(Z,1)
3630 PRINT AT 8,21;
3680 REM
3690 PRINT AT 4,0;
3730 PRINT AT 9,0;
3760 PRINT AT 11,0;
3770 PRINT "      ":REM 10 ESPACIOS
3780 PRINT AT 11,0;
3790 INPUT LINE "=="> "N$(Z,2)
3810 PRINT AT 9,21;
3860 REM
3870 PRINT AT 11,0;
4010 REM
4080 REM
4090 PRINT AT 11,0;
4150 PRINT AT 14,0;
4200 REM
4230 PRINT AT Z-1,0;
4240 FOR Q=1 TO 40:PRINT " ";:NEXT Q
4260 PRINT AT 4,0;
4310 PRINT AT 11,0;
4330 INPUT LINE "=="> "N$(Z,3)
4370 PRINT AT Z-1,0;
4380 FOR Q=1 TO 40:PRINT " ";:NEXT Q
4400 PRINT AT 4,0;
4440 PRINT AT 19,0;
4510 PRINT AT Z-1,0;
4520 FOR Q=1 TO 40:PRINT " ";:NEXT Q
4540 PRINT AT 4,0;
4580 PRINT AT 11,0;
4590 INPUT LINE "=="> "N$(1,4)
4740 PRINT AT Y,X;
4780 PRINT AT Y,X;
4800 PRINT AT Y,X;
4830 IF A$=CHR$(8) AND L0>0 THEN
      LET L0=L0-1:LET D$=D$ ( TO

```

TRUCOS Y RUTINAS BASICAS

```

      LO):LET X=X-1:PRINT AT
      Y,X;".":AT Y,X;:GOTO 4810
4890 PRINT AT Y,X;
4930 PRINT AT Y,X;
5050 PRINT AT Y,X;
5070 PRINT AT T,X=19;
5080 PRINT PAPER 0; " "
5140 PRINT AT Y,X+19;
5220 PRINT AT Y,X;
5230 FOR Q=1 TO L+3:PRINT " ";:NEXT
      Q
5280 PRINT AT Y,X+19;
5320 PRINT AT Y,X+19;
5390 PRINT AT Y,X+19;
5510 BEEP .2,15
5530 PRINT AT Y,X;
5580 PRINT AT Y,X;
5590 FOR Q=1 TO L:PRINT " ";:NEXT Q
5650 PRINT AT Y,X;
5660 FOR Q=1 TO L:PRINT " ";:NEXT Q
5750 IF F$(Z,1)<> CHR$(253) THEN
      GOTO 5810
6030 PRINT AT 0,X;
6050 PRINT AT 1,X;
6160 PRINT AT 2*Z+2,X;
6400 GOTO 6490*(SW=0)+
      6580*(SW=1)+6770*(SW=2)
6700 FOR X=1 TO LEN(F$(Z)):IF
      F$(Z,X)=CHR$(254) THEN LET
      CD=CD+1:IF CD=NC THEN LET
      Z1=X+1:LET X=LEN(F$(Z))
6710 NEXT X:FOR X=Z1 TO
      LEN(F$(Z)):IF
      J$(X-Z1+1)=F$(Z,X) THEN LET
      CR=CR+1:IF CR=LEN(J$) THEN
      LET X=LEN(F$(Z))
6810 IF F$(Z,1)<>CHR$(253) THEN
      GOTO 6870
6820 LET F$(Z)=F$(Z, 2 TO)
7130 FOR X=1 TO Z:FOR R=1 TO
      LEN(F$(Z)):IF F$(Z,R)=CHR$( 254)
      THEN LET CC=CC+1:IF CC=NC
      THEN LET Z1=R+1:LET R=L
      EN(F$(Z))
7140 NEXT R:LET CC=0:FOR R=1 TO
      LEN(F$(X)):IF F$(X,R)=C
      CHR$(254) THEN LET
      CC=CC+1:IF CC=NC THEN LET
      Z2=R+1:LET R=LEN(F$(X))
7150 NEXT R:IF F$(Z,Z1 TO)<F$(X,Z2
      TO) THEN LET A$=F$(Z):LET
      F$(Z)=F$(X):LET F$(X)=A$
7420 PRINT AT 8,0;
7450 DIM M$(N(1)+1,20)
7480 PRINT AT 15,0;
7500 PRINT AT 17,0;

```

```

7520 PRINT AT 15,0;
7550 PRINT AT 17,0;
7800 LOAD B$ DATA N()
7810 LOAD B$ DATA M$()
7820 REM
7830 REM
7840 REM
7850 REM
7860 REM
7870 REM
7880 REM
7890 REM
7900 CLS
8100 PRINT AT 9,0;
8170 PRINT AT 19,0;
8230 PRINT AT 9,0;
8240 FOR Q=1 TO 80:PRINT " ";:NEXT Q
8250 PRINT AT 19,0
8260 FOR Q=1 TO 80:PRINT " ";:NEXT Q
8370 IF F$(NF,1)=CHR$(253) THEN
      PRINT "ESTA FICHA ESTA
      BORRADA":GOSUB 5440:RETURN
8390 IF F$(Z,1)=CHR$(254) THEN LET
      CC=CC+1:PRINT:PRINT
      M$(CC);".":GOSUB 8410
8400 PRINT F$(NF,Z);

```

MSX y AMSTRAD:

```

      590 LOCATE 1,20
      2740 LOCATE 1,5
      4370 LOCATE 1,5
      3490 LOCATE 1,9
      3580 LOCATE 1,12
      3600 LOCATE 1,12
      3630 LOCATE 22,9
      3690 LOCATE 1,5
      3730 LOCATE 1,10
      3760 LOCATE 1,12
      3780 LOCATE 1,12
      3810 LOCATE 22,10
      3870 LOCATE 1,12
      4090 LOCATE 1,12
      4150 LOCATE 1,15
      4230 LOCATE 1,Z
      4260 LOCATE 1,5
      4310 LOCATE 1,12
      4370 LOCATE 1,Z
      4400 LOCATE 1,5
      4440 LOCATE 1,20
      4510 LOCATE 1,Z
      4540 LOCATE 1,5
      4580 LOCATE 1,12
      4740 LOCATE X,Y
      4780 LOCATE X,Y
      4800 LOCATE X,Y

```

```

4830 IF A$=CHR$(8) AND L0>0 THEN
      LET L0=L0-1:LET
      D$=LEFT$(D$,L0):LET X=X-1:LO-
      CATE X,Y:PRINT "-":LOCATE
      X,Y:GOTO 4810
4890 LOCATE X,Y
4930 LOCATE X,Y
5050 LOCATE X,Y
5070 LOCATE X+19,Y
5140 LOCATE X+19,Y
5220 LOCATE X,Y
5280 LOCATE X+19,Y
5320 LOCATE X+19,Y
5390 LOCATE X+19,Y
5530 LOCATE X,Y
5580 LOCATE X,Y
5650 LOCATE X,Y
6030 LOCATE X,1
6050 LOCATE X,2
6160 LOCATE X,2*Z+3

```

```

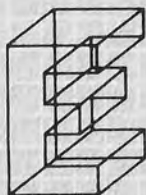
7420 LOCATE 1,9
7480 LOCATE 1,16
7500 LOCATE 1,18
7520 LOCATE 1,16
7550 LOCATE 1,18
8100 LOCATE 1,10
8170 LOCATE 1,20
8230 LOCATE 1,10
8250 LOCATE 1,20

```

Aunque las modificaciones que hay que realizar son numerosas, merece la pena introducir las para ver cómo se ejecuta el programa.

Debido a la gran extensión que han ocupado dichas modificaciones, se deja la explicación del programa para el próximo tomo. De momento podéis ir estudiando cómo funciona el programa general y los cambios que se han introducido en algunas de las rutinas.

Medidas de tiempo



El ordenador personal puede emplearse de muy diferentes maneras para medir el tiempo transcurrido entre dos sucesos. Al ser una máquina regida por un reloj muy estable, controlado generalmente por un cristal de

cuarzo, permite realizar medidas muy precisas asociadas al momento en que se producen.

Los lenguajes de programación más comunes incluyen sentencias para el empleo de la hora y la fecha dentro del programa, mediante la lectura de posiciones internas de la máquina, que se actualizan al transcurrir un determinado período. La señal física que marca el momento en que debe realizarse la actualización del contador interno es un oscilador o un circuito relacionado con el oscilador principal del sistema.

Vamos a describir algunos circuitos que permiten la generación de una frecuencia de referencia estable y diferentes formas de realizar medidas de intervalos de tiempo mediante ordenador personal. También se describirán circuitos que pueden añadirse al ordenador para poder efectuar mediciones que se salen del alcance de las posibilidades de la máquina.

Osciladores

Un circuito oscilador es aquel que genera una señal variable periódicamente con el tiempo. Una forma muy sencilla de oscilador puede realizarse con dos transistores conec-

tados en amplificador con emisor común y con la salida de cada uno conectada a la entrada del transistor opuesto a través de un circuito de temporización RC. La carga y descarga de los condensadores origina que los transistores pasen alternativamente de saturación a corte según las tensiones que se presenten a sus entradas. La figura muestra el circuito y las señales en los puntos principales.

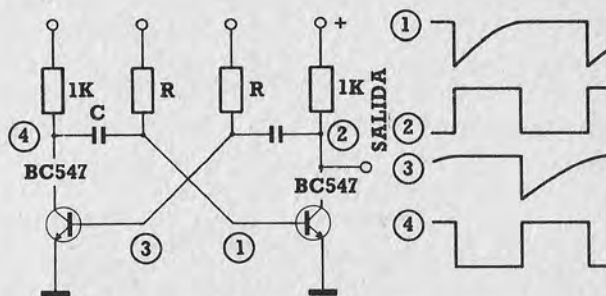


Fig. 1. Oscilador con transistores.

La señal generada por el oscilador es binaria y su frecuencia viene fijada por las constantes de tiempo definidas por R1 y C1 según la fórmula:

$$F = 2,2 / RC$$

La precisión conseguida con este montaje no es muy grande, pues depende de los valores de los componentes pasivos y del punto de conmutación de los transistores, por lo que variará con la temperatura.

El circuito puede simplificarse utilizando componentes digitales como se muestra en la figura, en el que los transistores han sido sustituidos por puertas inversoras. Para que las puertas se comporten como amplificadores es necesario "linealizarlas" mediante la conexión

EL TALLER DEL HARDWARE

en paralelo con la entrada y la salida de unas resistencias.

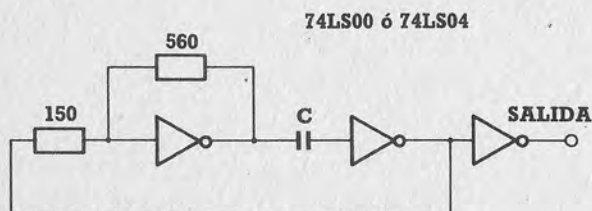


Fig. 2. Oscilador con puertas lógicas.

Para hacer la frecuencia más estable e independiente de los parámetros se puede diseñar un circuito en el que el parámetro fundamental sea dependiente de un elemento más estable como puede ser un cristal de cuarzo, convenientemente seleccionado. Con el circuito de la figura puede realizarse un oscilador con la estabilidad y precisión propia del cristal de cuarzo. El circuito integrado que se utiliza incluye internamente un oscilador controlado por tensión, que podría, por ejemplo, utilizarse para hacer una señal modulada en frecuencia a partir de una señal analógica, entrando por las entradas de CONTROL o de RANGO.

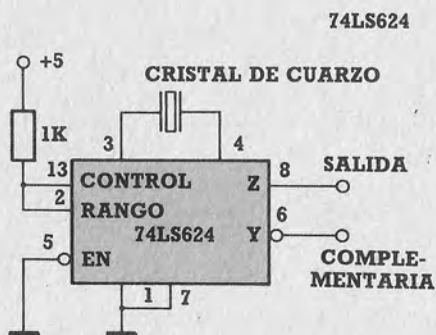


Fig. 3. Oscilador con cristal.

Por último podemos también utilizar como oscilador para base de tiempos de nuestras medidas el circuito ya utilizado en varios montajes anteriores: LM555, que tiene la limitación de frecuencia máxima de oscilación de 1 MHz.

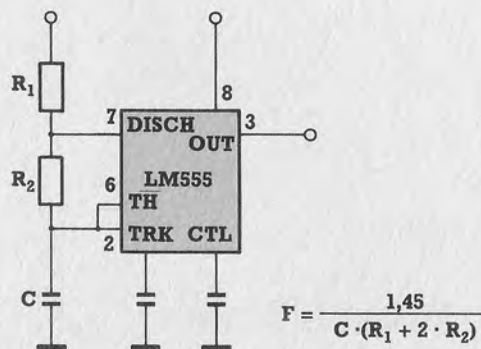


Fig. 4. Oscilador con LM555.

■ Circuito de cuenta

Una vez que disponemos de una base de tiempos estable y precisa, la medida de intervalos o períodos de tiempo requiere contar el número de veces que la señal de referencia se repite en dicho intervalo. Para ello hemos de montar un circuito que realice la cuenta de la frecuencia de base o de un divisor conocido, controlado por la señal a medir y por el ordenador. El circuito deberá tener como elemento fundamental un contador. En la figura vemos un contador con la entrada programable y que puede ser leído por el ordenador. También puede ser puesto a cero por programa.

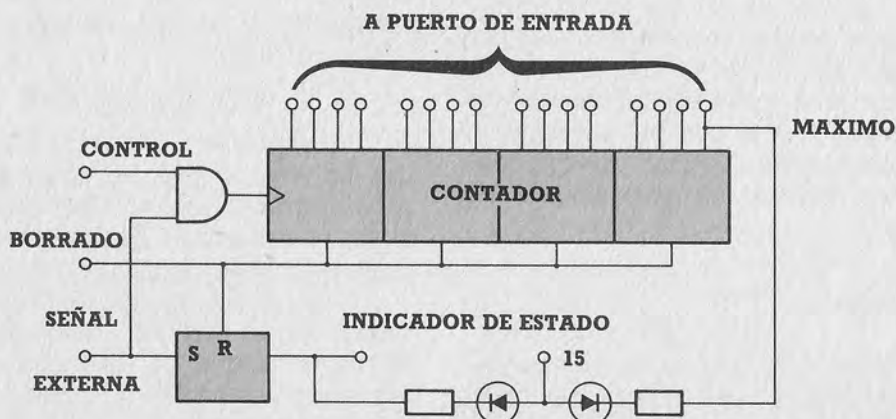


Fig. 5. Contador controlado por programa.

Para que desde el ordenador sepamos cuándo la señal a medir ha terminado, deberemos poder leer un indicador de estado, que puede ser un biestable activado por la señal de entrada, o bien una de las salidas del contador. Si el período a medir es muy largo, podríamos utilizar el contador externo como cuenta previa y un contador interno, que puede ser todo lo grande que necesitemos, para la cuenta total. Para determinar el valor del período medido deberemos multiplicar la cuenta del contador interno por el valor de la cuenta más larga del contador externo y sumarle el contenido del último período antes de finalizar la medida. Así podemos tener mucha resolución y un posible intervalo de medida ilimitado a un coste muy reducido. El programa, durante el período de lectura deberá estar observando el indicador de cuenta máxima para incrementar el contador interno cada vez que sobrepase el límite.

Algunos sucesos a medir pueden presentar problemas de cómo determinar en qué momento se debe empezar la lectura y en qué momento se debe terminar. Es necesario poder determinar las condiciones de disparo de forma clara, con circuito de filtro externo para separarla del resto de las señales. Por ejemplo, si es un fenómeno que presenta variaciones lentas, será necesario establecer el nivel a partir del cual la señal se considera en estado de lectura. Un amplificador de entrada con circuito de histéresis, los indicados en el tomo de lecturas analógicas, nos ayudará a "limpiar" las condiciones de comienzo y final. Para presentar de manera evidente en qué momento estamos contando, se dispondrá una señal luminosa, con diodo LED, para indicar que la señal supera el nivel y que, por tanto, la lectura sería válida. Además, para calibrar la lectura, podemos hacer una comparación con un cronómetro manual, que después de la calibración no será necesario.

Este es un ejemplo en que volvemos a ver la conveniencia de utilizar los circuitos electrónicos a través de programas, para hacerlos más eficaces y adaptables a nuestras necesidades. Los circuitos mostrados junto con los programas para activarlos, nos permitirán realizar muchos experimentos con medida precisa de intervalos de tiempo, para determinación de velocidades, distancias y cualquier otro fenómeno físico cuya manifestación sea una magnitud medible como intervalo de tiempo.

■ Circuitos de temporización propios de los ordenadores

Todos los ordenadores personales disponen de circuitería interna que asociada a las sentencias de BASIC permite realizar la medida "grosera" de intervalos de tiempo.

Cuando los tiempos a medir son del orden de décimas de segundo, la resolución del reloj de tiempo real incluido en los ordenadores como el IBM-PC, derivado del oscilador principal puede ser suficiente. Hay que tener presente el tiempo dedicado al refresco de la memoria dinámica y a la atención de las interrupciones de otros dispositivos, por lo que no puede garantizarse con dicho reloj la respuesta con resolución menor que la indicada. Mediante programas en BASIC puede realizarse medidas de tiempo, utilizando la sentencia `TIMER`, que da el valor del contador de tiempo interno, con resolución de 54,9 milisegundos y `TIME$` que da la hora, minuto y segundo. La sentencia `ON TIMER(M) GOSUB NNN` origina el comienzo de la ejecución de una rutina cada M segundos. En realidad lo que hace el BASIC es utilizar una posibilidad del sistema accesible desde cualquier lenguaje. Por cada interrupción de `TIMER0` se produce una entrada en la rutina apuntada por el vector número `X'1C'` y que es utilizable por el usuario. En esta rutina puede establecerse cualquier tipo de contabilización de tiempo con la resolución de 54,9 milisegundos o 18,2 pulsos por segundo.

Existe la posibilidad de realizar medida de tiempos con mayor resolución, utilizando los contadores internos del temporizador programable número 0. La señal de 18,2 pulsos por segundo está producida por el contador programable `TIMER0` que es activado por una señal derivada del reloj principal y que es de frecuencia 1,19 MHz. Mediante la programación del `TIMER0` se establece una carga inicial de 65.535 (`X'FFFF'`) que será decremada por cada pulso externo y producirá una entrada de interrupción al llegar a cero, cargándose otra vez con el valor inicial y empezando de nuevo la cuenta hacia abajo. La lectura del contador interno puede hacerse a través del puerto `X'40'` mediante dos IN seguidas, pues los 16 bits del contador han de leerse primero los 8 bits inferiores y luego los superiores. Para que la lectura sea válida debe hacerse con una rutina en lenguaje de máquina, pues si se hiciera con instrucciones del in-

EL TALLER DEL HARDWARE

térprete de BASIC, podrían resultar valores correspondientes a períodos diferentes. Con la rutina que se muestra puede realizarse la lectura de dicho contador interno mediante rutina en lenguaje de máquina. Pueden conseguirse resoluciones del orden de microsegundos, aunque es necesario tener en cuenta que no será posible medir dos sucesos muy próximos con esa resolución, al ser necesario un tiempo para el registro del valor de los contadores. La resolución será de microsegundos para sucesos separados por lo menos 100 microsegundos.

Rutina en lenguaje de máquina para medir tiempos

trica o el cierre de un circuito, puede realizarse utilizando las entradas de pulsadores de la tarjeta del adaptador de juegos. En el segundo programa de ejemplo puede verse cómo mediante las sentencias STRIG puede recibirse control, al producirse el cierre de cualquiera de los circuitos conectados a las entradas de los pulsadores. Si la resolución necesaria en la medida de tiempos es superior a los 54,9 milisegundos, podrá utilizarse la técnica indicada en el párrafo anterior, explorando las entradas mediante sentencias INP y marcando el tiempo en el que se produce el cierre del contacto.

En el programa de ejemplo se utiliza la

```

55      PUSH BP          GUARDA BP
8B EC   MOV BP,SP        APUNTA AL STACK CON BP
8B 7E 06 MOV DI,[BP+6]    TOMA DIRECCION DEL PARAMETRO
E4 40   IN AL,H040        LEE TIMER
8A E0   MOV AH,AL
E4 40   IN AL,H040
8B 25   MOV [DI],AH       RELLENA PARAMETRO
47      INC DI
8B 05   MOV [DI],AL       Y SIGUIENTES
89 05   MOV [DI],AX
8B 7E 08 MOV DI,[BP+8]
1E      PUSH DS
8B 40 00 MOV AX,H400
8E D8   MOV DS,AX
A1 6C 00 MOV AX,TIMER_LOW
1F      POP DS
89 05   MOV [DI],AX
5D      POP BP
CA 04   RET 4

```

Para conseguir la máxima resolución es conveniente pasar a modo 3 el temporizador. Se puede realizar con la siguiente rutina en código de máquina:

Rutina de CAMBIO DE MODO

rutina en lenguaje de máquina para acceder a los registros internos del temporizador sin afectar, en lo posible, al tiempo del período medido. En cada llamada a la rutina de registro del tiempo se almacena el contenido actual

```

FA      CLI              INHIBE INTERRUPCIONES
B0 36   MOV AL,36H       PONE VALOR DEL NUEVO MODO
E6 43   OUT TIMER+3      LO SACA AL 8253
B0 00   MOV AL,0         PONE A CERO EL CONTADOR
E6 40   OUT TIMER
E6 40   OUT TIMER
FB      STI              ADMITE INTERRUPCIONES DE NUEVO
CA 00   RET 0

```

Para medir intervalos entre sucesos externos, accesibles a través de una señal eléc-

del contador en la dirección de las variables pasadas como parámetros.

Programa de medida de tiempos para IBM-PC

```
10 REM MEDIDA DE TIEMPOS CON EL TIMER INTERNO
20 REM LECTURA DE TIMER Y DE LA POSICION X'46C'
30 REM SE MANTIENE TIMER EN MODO 2. IBM-PC
40 DEFINT A-Z
50 DIM A(100),B(100)
60 DEF SEG=&H1700:T=0
70 DATA &H55,&H8B,&HEC,&H8B,&H7E,&H06,&HE4,&H40
80 DATA &H8A,&HE0,&HE4,&H40,&H8B,&H25,&H47,&H8B
90 DATA &H05,&H8B,&H7E,&H0B,&H1E,&H8B,&H40,&H00
100 DATA &H8E,&HD8,&HA1,&H6C,&H00,&H1F,&H89,&H05
110 DATA &H5D,&HCA,&H04,&H00,&H00,&H00,&H00
120 FOR I=0 TO 35: READ V: POKE(I),V: NEXT I
130 FOR I=1 TO 20: CALL T(A(I),B(I)): NEXT I
140 CLS:PRINT "PULSOS    TIMER BAJO"
150 FOR I=1 TO 20
160 PRINT A(I);" ";HEX$(B(I))
170 NEXT I
```

En el programa de ejemplo se utiliza la sentencia TIMER para la medida del intervalo entre dos sucesos definidos por las entradas de los pulsadores del adaptador de juegos.

ploración del teclado, para realizar la medida de tiempos con resolución de 20 milisegundos, pues la INT se produce con una frecuencia de 50 Hz. Se muestra un ejemplo de programa

Programa de MEDIDA DE TIEMPOS con STRIG para IBMPC

```
10 REM MEDIDA DE TIEMPOS con STRIG
20 REM PROGRAMA DE MEDIDA DE INTERVALOS DE TIEMPO
30 REM POR LECTURA DE LAS ENTRADAS DE PULSADORES
40 REM A TRAVES DE LA TARJETA DE ADAPTADOR DE JUEGOS
50 REM IBM-PC
60 ON STRIG(0) GOSUB 150
70 ON STRIG(4) GOSUB 220
80 STRIG(0) ON: STRIG(4) ON
90 A=0: E$="ERROR EN SECUENCIA DE PULSOS"
100 FOR I=1 TO 10000:" Programa principal
110 IF A=2 THEN END
120 NEXT I
130 IF A=0 THEN PRINT "TIEMPO DE ESPERA TERMINADO"
140 END
150 REM COMIENZO DE LECTURA DE INTERVALO
160 IF A<>0 THEN RETURN
170 T1=TIMER
180 PRINT "COMIENZO LECTURA"
190 A=1
200 RETURN
210 REM FIN DE LECTURA DE INTERVALO
220 STRIG(4) OFF
230 IF A=1 THEN A=2 ELSE PRINT E$:STRIG(4) ON:RETURN
240 T2=TIMER
250 A=2
260 PRINT "FIN DE LECTURA"
270 TIEMPO=T2-T1
280 PRINT "TIEMPO TOTAL=";TIEMPO;" SEGUNDOS"
290 RETURN
```

En el SPECTRUM existe también la posibilidad de utilizar la temporización interna producida por la ULA y empleada para la ex-

que utiliza los valores internos de los contadores para efectuar medidas de tiempo.

Programa CRONOMETRO para Spectrum

```
10 REM CRONOMETRO SPECTRUM
20 PRINT AT 10,10:"PULSE UNA TECLA PARA EMPEZAR"
25 PAUSE: CLS
30 POKE 23672,0: POKE 23673,0: POKE 23674,0
40 PRINT #1:"PULSE 0 PARA PARAR"
50 PRINT AT 10,10:(65536)*(PEEK 23674)+255*
  (PEEK 23673)+PEEK 23672)/50
60 IF INKEY="0" THEN STOP
```

EL TALLER DEL HARDWARE

Puede utilizarse la interrupción de tiempo para activar acciones en momentos programados, sin necesidad de esperar las cuentas de los contadores internos. Para ello es necesario cambiar el modo de funcionamiento de las interrupciones a modo 2, con lo que se utiliza el registro interno I para obtener la dirección de la rutina a la que deberá pasarse el control al producirse la señal. La dirección se obtiene del valor de I (como parte alta de la dirección) multiplicado por 256, sumándole el valor leído del bus al producirse la interrupción. Esta dirección es en realidad el lugar de donde se extraerá la dirección de la rutina. Para usarlo se cargará en una zona de memoria reservada la rutina de tratamiento, en una palabra de 16 bits la dirección de la rutina y en el registro I (mediante un corto programa en lenguaje de máquina) la dirección de dicha palabra.

Una posibilidad más avanzada es la de utilizar un circuito de temporizador programable como los que ya posee el IBM-PC. Este circuito posee temporizadores que pueden programarse para realizar muy diferentes funciones. En el circuito que se muestra puede utilizarse para: medir número de ocurrencias de pulso, tiempo basado en un reloj de referencia entre dos pulsos y producir señales de frecuencia o duración programables.

Al SPECTRUM también podría conectarse una tarjeta con el mismo circuito 8253 o el Z80-CTC, direccionándolo de la forma acostumbrada.

■ Temporizador 8253

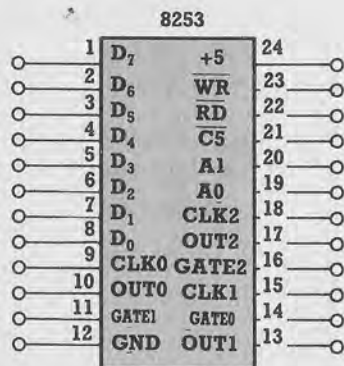


Fig. 6. Circuito 8253.

El circuito 8253 posee tres temporizadores programables. Cada uno de ellos tiene dos

entradas y una salida accesibles desde el exterior. De las entradas una es el reloj que fija el período de cuenta, siendo la otra entrada un control físico sobre el período de cuenta. La salida se activa según haya sido programado el temporizador, por ejemplo, para producir un pulso de salida cada vez que se hayan producido un determinado número de pulsos en la entrada. El contador interno es de 16 bits y puede programarse su valor inicial.

Dispone de seis modos de funcionamiento:

- Modo 0: Interrupción al final de la cuenta. Al inicializarse, la salida tomará valor bajo. Al alcanzarse el final de la cuenta, la salida tomará valor alto y lo mantendrá hasta que vuelva a recargarse de nuevo. Es un funcionamiento monoestable programado, de una sola vez y con interrupción.

- Modo 1: Un solo pulso. Al cargarse el valor de cuenta la salida toma valor alto. Al producirse un pulso de activación en la entrada, la salida toma valor bajo, que se mantendrá durante el período de cuenta.

- Modo 2: Divisor de frecuencia programable. La salida será una señal de frecuencia igual a la del reloj de entrada dividida por el valor de la cuenta programada. El contador interno se carga a su valor programado, cada vez que alcanza valor cero.

- Modo 3: Generador de señal cuadrada. Es muy similar al modo 2, con la diferencia de que el contador cuenta cada pulso de entrada como 2.

- Modo 4: Generador de pulso activado por programa. La salida presentará un pulso de nivel bajo al alcanzarse la cuenta programada y continuará de nuevo.

- Modo 5: Generador de pulso activado exteriormente. Se produce un pulso de la duración programada como consecuencia de un pulso en la entrada.

El temporizador 8253 que utiliza el IBM-PC está accesible en las siguientes direcciones de puertos:

- X'40': Contador 0. Carga de valor inicial o lectura.

- X'41': Contador 1. Carga de valor inicial o lectura.

- X'42': Contador 2. Carga de valor inicial o lectura.

- X'43': Control. Para inicialización o modificación de modos.

Ejemplo de aplicación

El circuito indicado en la figura utilizando el temporizador 8253-5 puede conectarse directamente al conector de expansión como un periférico bidireccional. La base de tiempos determinará la resolución de las medidas. Los circuitos de condiciones de las señales de entrada fijan los períodos de cuenta. Cada señal de condicionamiento de una señal externa deberá también ser accesible al ordenador para poder saber cuándo se ha cumplido la condición. Todos los valores de temporización interna son programables y, por tanto, pueden adaptarse a cualquier tipo de aplicación. Los temporizadores pueden utilizarse en cadena, con lo que pueden conseguirse mayores valores de cuenta o de duración de períodos programables. La señal de selección SEL deberá producirse como en los otros circuitos presentados mediante decodificadores, para ubicarlo en una zona de puertos no utilizada por otros adaptadores. Como el circuito tiene internamente 4 direcciones de puerto las entradas A0

y A1 seleccionan cada registro interno y la entrada CS selecciona el circuito.

El circuito se utiliza frecuentemente para producir señales de frecuencia constante durante un intervalo determinado. Uno de los temporizadores genera la señal, mientras otro mide la duración del intervalo. Al terminar la generación de una frecuencia, se programa la siguiente y la duración en el otro temporizador.

Conclusión

Con el ordenador personal y unos pocos circuitos adicionales podemos realizar medidas muy precisas de intervalos de tiempo entre sucesos, siempre que podamos obtener una señal eléctrica que permita indicar al contador cuándo tiene que contar los pulsos de una base de tiempos estable. Utilizando la medida precisa de tiempo podremos realizar numerosas medidas de fenómenos físicos que puedan generar señales convertibles en eléctricas y así aprovechar la potencia de nuestra máquina en aplicaciones realmente serias.

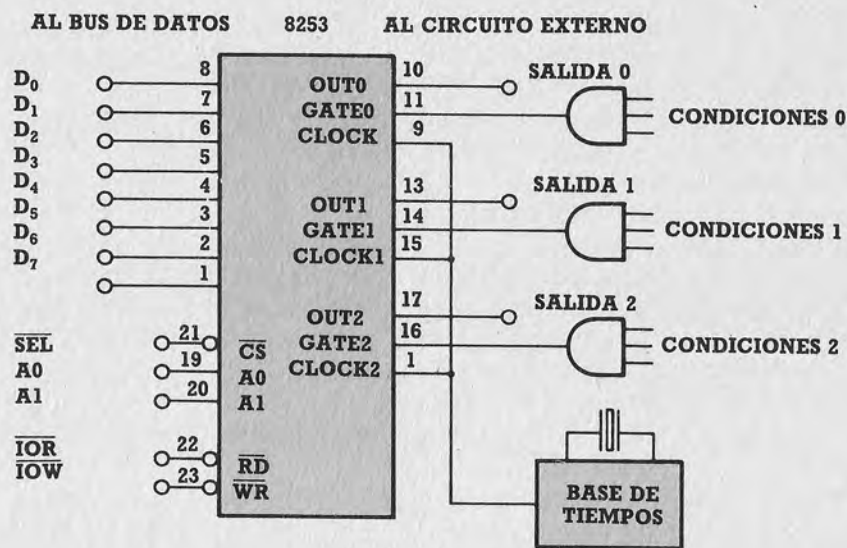
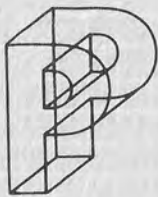


Fig. 7. Circuito de medida de intervalos con 8253.

NATURALEZA Y TECNOLOGIA

Genética



ARA enunciar de una manera general la importancia de la Genética podríamos decir que todos y cada uno de los caracteres físicos y psíquicos de un individuo vienen determinados por la herencia genética de sus antepasados y por la influencia del medio ambiente.

Cada carácter está determinado por dos genes, uno procedente del padre y otro de la madre (que a su vez recibieron los suyos de sus padres, y así sucesivamente), que se llaman generación **P** o paterna, y que serán los encargados de dotar a los descendientes, que se llaman generación **F** o filial, de ciertas cualidades, y de hacer que este individuo sea portador de otras muchas que pueden estar ocultas y transmitirlas a sus descendientes.

El comportamiento de la transmisión de

caracteres puede estudiarse teniendo en cuenta un carácter (monohíbridos), o varios caracteres (polihíbridos). Nuestro programa tratará del primer caso. Mendel realizó ensayos basados en todo lo anterior y elaboró las famosas tres leyes que llevan su nombre.

Hay genes dominantes y genes recesivos. Un gen dominante es aquél que se manifiesta en el fenotipo y uno recesivo es aquél que no se manifiesta, diciéndose, por tanto, que está oculto. Los genes se representan con letras, el dominante con mayúscula y el recesivo con la misma letra, pero en minúscula. La letra que se escoge suele ser la inicial del carácter dominante.

En este programa se selecciona aleatoriamente una determinada especie con el carácter correspondiente, indicando asimismo cuál es la alternativa dominante y cuál la recesiva y a continuación también aleatoriamente se elige el padre y la madre. La clave estará en hallar el porcentaje de uno y otro carácter que tendrá el fenotipo de los descendientes.

```
1 SCREEN 2
10 REM *****
20 REM * PROBLEMAS DE GENETICA *
30 REM *****
40 REM ***** INSTRUCCIONES *****
45 CLS
50 PRINT "***** I N S T R U C C I O N E S *****"
*****"
60 PRINT:PRINT:PRINT:PRINT
70 PRINT "ESTAS EN UN LABORATORIO DE GENETICA Y TU AYUDANTE"
80 PRINT "TE PLANTEA UN PROBLEMA DE CRUZAMIENTO DE UNA DETERMINADA ESPECIE."
90 PRINT "LOS GENES DE ESTA ESPECIE DETERMINARAN UN CARACTER PROPIO"
100 PRINT "QUE TENDRA DOS ALTERNATIVAS: UNA DOMINANTE Y OTRA RECESIVA."
110 PRINT
120 PRINT "BASANDOTE EN LAS CARACTERISTICAS GENETICAS DEL PADRE Y DE LA MADRE"
130 PRINT "TENDRAS QUE HALLAR EL PORCENTAJE DE GENES DESCENDIENTES"
140 PRINT "DE UNA Y OTRA ALTERNATIVA."
```

APRENDER CON EL ORDENADOR

```

150 PRINT:PRINT:PRINT
160 PRINT "                                *PULSA UNA TECLA PARA CONTINUAR* "
170 LET P$=INKEY$: IF P$="" THEN GOTO 170
180 REM ***** LECTURA DATOS DEL PROBLEMA *****
185 CLS
190 DIM A$(80):DIM B$(20,4):DIM C$(20):DIM D$(20):DIM H$(4)
200 FOR I=1 TO 80
210 READ A$(I)
220 NEXT I
230 FOR I= 1 TO 20
240 FOR J=1 TO 4
250 READ B$(I,J)
260 NEXT J
270 NEXT I
280 FOR I=1 TO 20
290 READ C$(I)
300 NEXT I
310 FOR I=1 TO 20
320 READ D$(I)
330 NEXT I
340 REM *****ENUNCIADO PROBLEMA *****
350 RANDOMIZE TIMER
360 LET J=INT(RND*20)+1
370 PRINT "ESPECIE:";A$(J)
380 PRINT
390 PRINT "CARACTER:";A$(J+20)
400 PRINT
410 PRINT "ALTERNATIVA:";A$(2*J+39);"/";A$(2*J+40)
420 PRINT
430 PRINT "GEN DOMINANTE:";A$(2*J+39);" (";B$(J,1);")"
440 PRINT
450 PRINT "GEN RECESIVO:";A$(2*J+40);" (";B$(J,3);")"
460 PRINT: PRINT "                                ****INDIVIDUOS A CRUZAR****"
470 PRINT:PRINT
480 FOR I=1 TO 4
490 LET X=INT(RND*4)+1
500 LET H$(I)=B$(J,X)
510 NEXT I
520 PRINT "PADRE:";H$(1);H$(2)
530 PRINT "MADRE:";H$(3);H$(4)
540 PRINT:PRINT
550 PRINT "PORCENTAJE DE DESCENDIENTES ";C$(J):INPUT X1
560 PRINT "PORCENTAJE DE DESCENDIENTES ";D$(J):INPUT X2
570 REM ***** SOLUCION *****
580 LET Z=0
590 FOR I=1 TO 2
600 IF ASC(H$(I))<97 OR ASC(H$(3)) < 97 THEN Z=Z+1
610 IF ASC(H$(I))<97 OR ASC(H$(4)) < 97 THEN Z=Z+1
620 NEXT I
630 LET Y=(Z*100)/4
640 IF X1=Y AND X2=(100-Y) THEN PRINT "RESPUESTA CORRECTA"
650 IF X1<>Y OR X2<>(100-Y) THEN PRINT "RESPUESTA INCORRECTA"
660 PRINT "PULSE UNA TECLA PARA VER PROBLEMA RESUELTO"
670 LET P$=INKEY$:IF P$="" THEN GOTO 670
680 CLS
690 PRINT "                                PADRE                                MADRE"
695 PRINT
700 PRINT "                                ";H$(1);H$(2);"                                ";H$(3)
705 PRINT:PRINT:PRINT
710 PRINT "                                ";H$(1);"                                ";H$(2);"                                ";H$(3);"
715 PRINT "                                ";H$(4)
720 PRINT :PRINT :PRINT :PRINT :PRINT :PRINT :PRINT :PRINT :PRINT :PRINT :PRINT
730 PRINT "                                ";H$(1);H$(3);"                                ";H$(2);H$(3);"                                ";
H$(2);H$(4);"                                ";H$(1);H$(4)
739 LET A$="M+0,+110"
740 PSET(55,100):DRAW A$
741 LET A$="M+479,+110"
742 PSET(55,100):DRAW A$
743 LET A$="M-313,+110"
744 PSET(372,100):DRAW A$
745 LET A$="M+0,+110"
746 PSET(534,100):DRAW A$
747 LET A$="M-150,+110"
748 PSET(370,100):DRAW A$
749 LET A$="M+0,+110"
750 PSET(220,100):DRAW A$
751 LET A$="M+150,+110"
752 PSET(220,100):DRAW A$
753 LET A$="M-150,+110"

```

```

754 PSET(534,100):DRAW A$
1000 REM ***** FINAL DEL PROGRAMA *****
1010 PRINT:PRINT "PULSE CUALQUIER TECLA PARA CONTINUAR"
1020 PRINT "S PARA SALIR DEL PROGRAMA"
1030 LET Z$=INKEY$:IF Z$=""THEN GOTO 1030
1040 IF Z$="S"OR Z$="s" THEN END
1050 CLS:GOTO 340
2000 DATA COBAYOS,HOMBRE,MOSCA,HOMBRE,HOMBRE,HOMBRE,HOMBRE,HOMBRE,HOMBRE,HOMBRE,
HOMBRE,OVEJA,GALLINA,CERDO,CONEJO,HOMBRE,TORO,PLANTA,MOSCA,CALABAZA DE VERANO
2010 DATA COLOR DEL PELO,SORDOMUDEZ,COLOR DEL CUERPO,COLOR DE LOS OJOS,COLOR DEL
PELO,TIPO DE PELO,DALTONISMO,ALBINISMO,HEMOFILIA,TIPO DE OJOS,FACTOR Rh,COLOR D
E LA LANA,TIPO DE CRESTA,TIPO DE OREJAS, COLOR DEL PELO,MIOPIA,CUERNOS
2020 DATA TIPO DE SEMILLA,FORMA DE LAS ALAS,COLOR
2030 DATA NEGRO,BLANCO,NORMAL,SORDOMUDO,GRIS,NEGRO,OSCURO,CLARO,OSCURO,RUBIO,RIZ
ADO,LISO,NORMAL,DALTONICO,NORMAL,ALBINO,NORMAL,HEMOFILICO,SESGADOS,NORMALES,Rh+,
Rh-,BLANCO,NEGRO,EN ROSA,ASERRADA,PELUDA,DESNUDA,NEGRO,LEONADO,MIOPE
2040 DATA VISION NORMAL,FALTA,PRESENCIA,LISA,RUGOSA,NORMAL,VESTIGIAL,BLANCO,AMAR
ILLO
2050 DATA N,N,n,n,N,N,n,G,G,g,g,O,O,o,o,O,O,o,o,R,R,r,r,N,N,n,n,N,N,n,n,N,N,n,
n,S,S,s,s,R,R,r,r,B,B,b,b,E,E,e,e,P,P,p,p,N,N,n,n,M,M,m,m,F,F,f,f,L,L,l,l,N,N,n,
n,B,B,b,b
2060 DATA DE PELO NEGRO,NORMALES,DE CUERPO GRIS,DE OJOS OSCUROS,DE PELO OSCURO,D
E PELO RIZADO,NORMALES,NORMALES,NORMALES,CON OJOS SESEGADOS,CON Rh+,DE LANA BLANC
A,DE CRESTA EN ROSA,DE OREJAS PELUDAS,DE PELO NEGRO,MIOPE,SIN CUERNOS
2070 DATA DE SEMILLA LISA,DE ALAS NORMALES,DE COLOR BLANCO
2080 DATA DE PELO BLANCO,SORDOMUDOS,DE CUERPO NEGRO,DE OJOS CLAROS,DE PELO RUBIO
,DE PELO LISO,DALTONICOS,ALBINOS,HEMOFILICOS,CON OJOS NORMALES,CON Rh-,DE LANA N
EGRA,DE CRESTA ASERRADA,DE OREJAS DESNUDAS,DE PELO LEONADO,NORMALES,CON CUERNOS
2090 DATA DE SEMILLA RUGOSA,DE ALAS VESTIGIALES,DE COLOR AMARILLO

1 SCREEN 2
10 REM *****
20 REM * PROBLEMAS DE GENETICA *
30 REM *****
40 REM ***** INSTRUCCIONES *****
45 CLS
50 PRINT "***** I N S T R U C C I O N E S *****"
*****
60 PRINT:PRINT:PRINT
70 PRINT "ESTAS EN UN LABORATORIO DE GENETICA Y TU AYUDANTE"
80 PRINT "TE PLANTEA UN PROBLEMA DE CRUZAMIENTO DE UNA DETERMINADA ESPECIE."
90 PRINT "LOS GENES DE ESTA ESPECIE DETERMINARAN UN CARACTER PROPIO"
100 PRINT "QUE TENDRA DOS ALTERNATIVAS: UNA DOMINANTE Y OTRA RECESIVA."
110 PRINT
120 PRINT "BASANDOTE EN LAS CARACTERISTICAS GENETICAS DEL PADRE Y DE LA MADRE"
130 PRINT "TENDRAS QUE HALLAR EL PORCENTAJE DE GENES DESCENDIENTES"
140 PRINT "DE UNA Y OTRA ALTERNATIVA."
150 PRINT:PRINT:PRINT
160 PRINT " *PULSA UNA TECLA PARA CONTINUAR* "
170 LET P$=INKEY$: IF P$="" THEN GOTO 170
180 REM ***** LECTURA DATOS DEL PROBLEMA *****
185 CLS
190 DIM A$(80):DIM B$(20,4):DIM C$(20):DIM D$(20):DIM H$(4)
200 FOR I=1 TO 80
210 READ A$(I)
220 NEXT I
230 FOR I= 1 TO 20
240 FOR J=1 TO 4
250 READ B$(I,J)
260 NEXT J
270 NEXT I
280 FOR I=1 TO 20
290 READ C$(I)
300 NEXT I
310 FOR I=1 TO 20
320 READ D$(I)
330 NEXT I
340 REM *****ENUNCIADO PROBLEMA *****
350 RANDOMIZE TIMER
360 LET J=INT(RND*20)+1
370 PRINT "ESPECIE:";A$(J)
380 PRINT
390 PRINT "CARACTER:";A$(J+20)
400 PRINT
410 PRINT "ALTERNATIVA:";A$(2*J+39);"/";A$(2*J+40)
420 PRINT
430 PRINT "GEN DOMINANTE:";A$(2*J+39);" (";B$(J,1);")"
440 PRINT
450 PRINT "GEN RECESIVO:";A$(2*J+40);" (";B$(J,3);")"
460 PRINT:PRINT " *****INDIVIDUOS A CRUZAR*****"

```

APRENDER CON EL ORDENADOR

```

470 PRINT:PRINT
480 FOR I=1 TO 4
490 LET X=INT(RND*4)+1
500 LET H$(I)=B$(J,X)
510 NEXT I
520 PRINT "PADRE:";H$(1);H$(2)
530 PRINT "MADRE:";H$(3);H$(4)
540 PRINT:PRINT
550 PRINT "PORCENTAJE DE DESCENDIENTES ";C$(J):INPUT X1
560 PRINT "PORCENTAJE DE DESCENDIENTES ";D$(J):INPUT X2
570 REM ***** SOLUCION *****
580 LET Z=0
590 FOR I=1 TO 2
600 IF ASC(H$(I))<97 OR ASC(H$(3)) < 97 THEN Z=Z+1
610 IF ASC(H$(I))<97 OR ASC(H$(4)) < 97 THEN Z=Z+1
620 NEXT I
630 LET Y=(Z*100)/4
640 IF X1=Y AND X2=(100-Y) THEN PRINT "RESPUESTA CORRECTA"
650 IF X1<>Y OR X2<>(100-Y) THEN PRINT "RESPUESTA INCORRECTA"
660 PRINT "PULSE UNA TECLA PARA VER PROBLEMA RESUELTO"
670 LET P$=INKEY$:IF P$="" THEN GOTO 670
680 CLS
690 PRINT "                PADRE                                MADRE"
695 PRINT
700 PRINT "                ";H$(1);H$(2);"                ";H$(3)
);H$(4)
705 PRINT:PRINT:PRINT
710 PRINT "                ";H$(1);"                ";H$(2);"                ";H$(3);"
";H$(4)
720 PRINT :PRINT :PRINT :PRINT :PRINT :PRINT :PRINT :PRINT
730 PRINT "                ";H$(1);H$(3);"                ";H$(2);H$(3);"                ";
H$(2);H$(4);"                ";H$(1);H$(4)
739 LET A$="M+0,+110"
740 PSET(55,100):DRAW A$
741 LET A$="M+479,+110"
742 PSET(55,100):DRAW A$
743 LET A$="M-313,+110"
744 PSET(372,100):DRAW A$
745 LET A$="M+0,+110"
746 PSET(534,100):DRAW A$
747 LET A$="M-150,+110"
748 PSET(370,100):DRAW A$
749 LET A$="M+0,+110"
750 PSET(220,100):DRAW A$
751 LET A$="M+150,+110"
752 PSET(220,100):DRAW A$
753 LET A$="M-150,+110"
754 PSET(534,100):DRAW A$
1000 REM ***** FINAL DEL PROGRAMA *****
1010 PRINT:PRINT "PULSE CUALQUIER TECLA PARA CONTINUAR"
1020 PRINT "S PARA SALIR DEL PROGRAMA"
1030 LET Z$=INKEY$:IF Z$="" THEN GOTO 1030
1040 IF Z$="S" OR Z$="s" THEN END
1050 CLS:GOTO 340
2000 DATA COBAYOS,HOMBRE,MOSCA,HOMBRE,HOMBRE,HOMBRE,HOMBRE,HOMBRE,HOMBRE,HOMBRE,
HOMBRE,OVEJA,GALLINA,CERDO,CONEJO,HOMBRE,TORO,PLANTA,MOSCA,CALABAZA DE VERANO
2010 DATA COLOR DEL PELO,SORDOMUDEZ,COLOR DEL CUERPO,COLOR DE LOS OJOS,COLOR DEL
PELO,TIPO DE PELO,DALTONISMO,ALBINISMO,HEMOFILIA,TIPO DE OJOS,FACTOR Rh,COLOR D
E LA LANA,TIPO DE CRESTA,TIPO DE OREJAS, COLOR DEL PELO,MIOPIA,CUERNOS
2020 DATA TIPO DE SEMILLA,FORMA DE LAS ALAS,COLOR
2030 DATA NEGRO,BLANCO,NORMAL,SORDOMUDO,GRIS,NEGRO,OSCURO,CLARO,OSCURO,RUBIO,RIZ
ADO,LISO,NORMAL,DALTONICO,NORMAL,ALBINO,NORMAL,HEMOFILICO,SESGADOS,NORMALES,Rh+,
Rh-,BLANCO,NEGRO,EN ROSA,ASERRADA,PELUDA,DESNUDA,NEGRO,LEONADO,MIOPE
2040 DATA VISION NORMAL,FALTA,PRESENCIA,LISA,RUGOSA,NORMAL,VESTIGIAL,BLANCO,AMAR
ILLO
2050 DATA N,N,n,n,n,n,n,n,G,G,g,g,o,o,o,o,R,R,r,r,N,N,n,n,n,n,n,n,n,n,
n,S,S,s,s,R,R,r,r,B,B,b,b,E,E,e,e,P,p,p,p,N,N,n,n,M,M,m,m,F,F,f,f,L,L,l,l,N,N,n,
n,B,B,b,b
2060 DATA DE PELO NEGRO,NORMALES,DE CUERPO GRIS,DE OJOS OSCUROS,DE PELO OSCURO,D
E PELO RIZADO,NORMALES,NORMALES,NORMALES,CON OJOS SESGADOS,CON Rh+,DE LANA BLANC
A,DE CRESTA EN ROSA,DE OREJAS PELUDAS,DE PELO NEGRO,MIOPE,SIN CUERNOS
2070 DATA DE SEMILLA LISA,DE ALAS NORMALES,DE COLOR BLANCO
2080 DATA DE PELO BLANCO,SORDOMUDOS,DE CUERPO NEGRO,DE OJOS CLAROS,DE PELO RUBIO
,DE PELO LISO,DALTONICOS,ALBINOS,HEMOFILICOS,CON OJOS NORMALES,CON Rh-,DE LANA N
EGRA,DE CRESTA ASERRADA,DE OREJAS DESNUDAS,DE PELO LEONADO,NORMALES,CON CUERNOS
2090 DATA DE SEMILLA RUGOSA,DE ALAS VESTIGIALES,DE COLOR AMARILLO

```

Modificaciones para otros equipos

El programa es válido para el IBM PC, siendo las variaciones para los demás ordenadores las siguientes:

AMSTRAD

1 MODE 1

Eliminar línea 350.

En las líneas 739-754:

— Utilizar MOVE(X,Y) en vez de PSET(X,Y).

— Ajustar coordenadas a la pantalla gráfica MODE 1 de AMSTRAD.

— Sustituir A\$ por A y suprimir M.

MSX

Las mismas que para Amstrad, excepto:

1 SCREEN 1

COMMODORE

45 PRINT CHR\$(147)

185 PRINT CHR\$(147)

680 PRINT CHR\$(147)

1050 PRINT CHR\$(147)

Eliminar línea 350.

En vez de líneas 739-754 utilizar subrutina de gráficos del tomo 1 (sección trucos de programación).

ZX-SPECTRUM

190 DIM A\$(20,18):DIM B\$(20,4):DIM

C\$(20,25):DIM D\$(20,25):DIM H\$(4,1)

350 RANDOMIZE 0

600 IF CODE H\$(1)<97 OR CODE H\$(3)<97 THEN Z=Z+1

610 IF CODE H\$(1)<97 OR CODE H\$(4)<97 THEN Z=Z+1

En las líneas 739-754:

— Utilizar PLOT (X,Y) en vez de PSET (X,Y).

— Ajustar coordenadas a la pantalla gráfica de alta resolución del Spectrum (256 columnas por 176 filas).

— Sustituir A\$ por A y suprimir M.
1040 IF Z\$="S" OR Z\$="s" THEN GOTO 9999

Todos los elementos de las sentencias DATA deberán ir entre comillas.

MATEMATICAS

■ Areas de figuras geométricas

No cabe duda de que la Geometría es una de las partes más importantes de la Matemática, sobre todo por la utilidad práctica que ésta tiene en la vida real.

¿A quién no le surge en algún momento la necesidad de forrar un libro, empapelar una pared, alicatar un suelo o incluso cercar un terreno?

Y todos estos son los ejemplos más sencillos que pueden presentarse, claro que, por ahora, sólo trataremos el tema de los polígonos regulares, sin entrar en otros planteamientos que resultarían más complicados; de todas maneras, ésta es la base de este tipo de Matemática práctica.

Estas fórmulas que, a veces, son difíciles de recordar, conviene que podamos tenerlas archivadas, para que en cualquier momento podamos disponer de ellas de una manera rápida y sencilla.

En el programa que a continuación presentamos se elegirá la opción deseada del menú, y después de dar los datos pertinentes para cada figura aparecerá el área de la figura elegida y un dibujo cualitativo de la misma.

```
1 SCREEN 2
10 CLS
20 REM*****PRESENTACION*****
30 PRINT "*****AREAS DE FIGURAS GEOMETRICAS*****"
40 PRINT:PRINT
50 PRINT TAB(7);"1 CUADRADO"
60 PRINT TAB(7);"2 RECTANGULO"
70 PRINT TAB(7);"3 TRIANGULO"
80 PRINT TAB(7);"4 ROMBO"
90 PRINT TAB(7);"5 ROMBOIDE"
100 PRINT TAB(7);"6 TRAPECIO"
110 PRINT TAB(7);"7 POLIGONOS DE MAS DE 4 LADOS"
120 PRINT TAB(7);"8 CIRCULO"
130 PRINT TAB(7);"9 SECTOR CIRCULAR"
140 PRINT TAB(7);"10 CORONA CIRCULAR"
```

APRENDER CON EL ORDENADOR

```

145 PRINT :PRINT :PRINT :PRINT
150 INPUT "PULSA EL NUMERO DE TU OPCION";A
160 CLS:PI=3.14159
170 ON A GOSUB 300,400,500,600,700,800,900,1000,1100,1200
180 GOTO 30
300 GOSUB 2000
310 INPUT "LONGITUD DEL LADO=";L
320 PRINT TAB(7);"AREA=";L^2
330 GOTO 3000
400 GOSUB 2100
410 INPUT "LONGITUD DE LA BASE=";B
420 INPUT "LONGITUD DE LA ALTURA=";H
430 PRINT TAB(7);"AREA=";B*H
440 GOTO 3000
500 GOSUB 2200
510 INPUT "LONGITUD DE LA BASE=";B
520 INPUT "LONGITUD DE LA ALTURA=";H
530 PRINT TAB(7);"AREA=";B*H/2
540 GOTO 3000
600 GOSUB 2300
610 INPUT "LONGITUD DE LA DIAGONAL MAYOR=";D1
620 INPUT "LONGITUD DE LA DIAGONAL MENOR=";D2
630 PRINT TAB(7);"AREA=";D1*D2/2
640 GOTO 3000
700 GOSUB 2400
710 INPUT "LONGITUD DE LA BASE=";B
720 INPUT "LONGITUD DE LA ALTURA=";H
730 PRINT TAB(7);"AREA=";B*H
740 GOTO 3000
800 GOSUB 2500
810 INPUT "LONGITUD DE LA BASE MAYOR=";B1
820 INPUT "LONGITUD DE LA BASE MENOR=";B2
830 INPUT "LONGITUD DE LA ALTURA=";H
840 PRINT TAB(7);"AREA=";(B1+B2)*H/2
850 GOTO 3000
900 CLS
905 PRINT "*****POLIGONOS REGULARES DE MAS DE CUATRO LADOS*****"
*****
910 PRINT :PRINT :PRINT :PRINT
920 PRINT TAB(7);"1 PENTAGONO":PRINT
930 PRINT TAB(7);"2 HEXAGONO":PRINT
940 PRINT TAB(7);"3 HEPTAGONO":PRINT
950 PRINT TAB(7);"4 OCTOGONO":PRINT
960 PRINT TAB(7);"5 ENEAGONO":PRINT
970 PRINT TAB(7);"6 DECAGONO":PRINT
975 INPUT "NUMERO DEL POLIGONO ELEGIDO=";N
980 IF N<1 OR N>6 THEN GOTO 900
985 CLS:INPUT "LONGITUD DEL LADO=";L
990 INPUT "LONGITUD DE LA APOTEMA=";A
995 PRINT TAB(7);"AREA DEL POLIGONO=";(N+4)*L*A/2
997 GOTO 3000
1000 GOSUB 2600
1010 INPUT "LONGITUD DEL RADIO=";R:PRINT
1020 PRINT TAB(7);"AREA=";PI*R*R
1030 GOTO 3000
1100 GOSUB 2700
1110 INPUT "LONGITUD DEL RADIO=";R:PRINT
1120 INPUT "ANGULO DEL SECTOR CIRCULAR (en grados)=";B
1130 PRINT TAB(7);"AREA=";PI*R*R*B/360
1140 GOTO 3000
1200 GOSUB 2800
1210 INPUT "LONGITUD DEL RADIO MAYOR=";R1:PRINT
1220 INPUT "LONGITUD DEL RADIO MENOR=";R2:PRINT
1230 PRINT TAB(7);"AREA=";(PI*R1*R1)-(PI*R2*R2)
1240 GOTO 3000
2000 PRINT "                                C U A D R A D O":A$="M+200,+0"
2005 PRINT:PRINT
2010 PSET (100,150):DRAW A$
2020 A$="M+0,-100":DRAW A$
2030 A$="M-200,+0":DRAW A$
2040 A$="M+0,+100":DRAW A$
2050 RETURN
2100 PRINT "                                R E C T A N G U L O":A$="M+200,+0"
2105 PRINT:PRINT
2110 PSET (100,150):DRAW A$
2120 A$="M+0,-50":DRAW A$
2130 A$="M-200,+0":DRAW A$
2140 A$="M+0,+50":DRAW A$
2150 RETURN
2200 PRINT "                                T R I A N G U L O":A$="M+200,+0"

```

```

2205 PRINT:PRINT
2210 PSET (100,150):DRAW A$
2220 A$="M-100,-100":DRAW A$
2230 A$="M-100,+100":DRAW A$
2240 RETURN
2300 PRINT "          R O M B O":A$="M+150,-50"
2305 PRINT:PRINT
2310 PSET (200,150):DRAW A$
2320 A$="M-150,-50":DRAW A$
2330 A$="M-150,+50":DRAW A$
2340 A$="M+150,+50":DRAW A$
2350 RETURN
2400 PRINT "          R O M B O I D E":A$="M+200,+0"
2405 PRINT:PRINT
2410 PSET (100,150):DRAW A$
2420 A$="M+100,-50":DRAW A$
2430 A$="M-200,+0":DRAW A$
2440 A$="M-100,+50":DRAW A$
2450 RETURN
2500 PRINT "          T R A P E C I O":A$="M+200,+0"
2505 PRINT:PRINT
2510 PSET (100,150):DRAW A$
2520 A$="M-50,-50":DRAW A$
2530 A$="M-100,+0":DRAW A$
2540 A$="M-50,+50":DRAW A$
2550 RETURN
2600 PRINT "          C I R C U L O"
2605 PRINT:PRINT
2610 CIRCLE (200,100),100,1,0,2*PI
2620 RETURN
2700 PRINT "          S E C T O R   C I R C U L A R"
2705 PRINT:PRINT
2710 CIRCLE (250,100),100,1,-2*PI,-PI/4
2720 RETURN
2800 PRINT "          C O R D O N A   C I R C U L A R"
2805 PRINT:PRINT
2810 CIRCLE (300,100),100,1,0,2*PI
2820 CIRCLE (300,100),50,1,0,2*PI
2830 RETURN
3000 REM *****final del programa *****
3010 LOCATE 22,5: PRINT "pulse S para salir"
3020 LET Z$=INKEY$:IF Z$=""THEN GOTO 3020
3030 IF Z$="s" OR Z$="S" THEN END
3040 GOTO 10

```

Modificaciones para otros equipos

SPECTRUM

Eliminar línea 1

```

160 CLS
170 LET Z=200+A*100
175 IF Z<300 OR Z>1200 THEN GOTO 10
180 GOSUB Z

```

Para dibujar los gráficos, sustituir PSET (X,Y) por PLOT (X,Y).

Eliminar variables A\$ y adecuar coordenadas a la pantalla del SPECTRUM.

```

2610 CIRCLE (127,80,50)
2710 PLOT (127,80):DRAW (50,0):DRAW
(-25,40,PI/4):DRAW (-25,-40)
2810 CIRCLE (127,80,50)
2820 CIRCLE (127,80,25)
3010 PRINT AT 20,5:"PULSE S PARA SALIR"
3030 IF Z$="S" OR Z$="s" THEN GOTO 9999

```

COMMODORE

Eliminar línea 1.

```

10 PRINT CHR$(147)
160 PRINT CHR$(147)
Para dibujar gráficos ver subrutina de dibujos
del tomo 1
900 PRINT CHR$(147)
3010 PRINT "PULSE S PARA SALIR"

```

AMSTRAD

1 MODE 1

Para dibujar gráficos utilizar PLOT en vez de PSET. Utilizar A\$ por A. Sustituir DRAW por DRAW R Y adecuar coordenadas a la pantalla gráfica MODE 1

MSX

Las mismas modificaciones que para AMSTRAD excepto:

1 SCREEN 1


```

480 LET W=INT(RND*25)+1
490 CLS
500 FOR N=1 TO 10
510 IF D(N)=W THEN GOTO 470
520 NEXT N
530 LET X=X+1
540 LET D(X)=W
550 PRINT A$(W)
560 PRINT
570 PRINT
580 PRINT
590 PRINT TAB(2);B$((3*W)-2)
600 PRINT
610 PRINT TAB(2);B$((3*W)-1)
620 PRINT
630 PRINT TAB(2);B$(3*W)
640 PRINT:PRINT
650 REM *****
660 REM ** RESPUESTAS **
670 REM *****
680 INPUT "RESPUESTA:";C
690 IF C=C(W) THEN GOSUB 740
700 IF C<>C(W) THEN GOSUB 820
710 CLS
720 IF X=10 THEN GOTO 900
730 GOTO 470
740 REM *****
750 REM ** RESPUESTA CORRECTA **
760 REM *****
770 PRINT
780 PRINT "      ¡CORRECTO!"
790 FOR Q=1 TO 1000:NEXT Q
800 LET A=A+1
810 RETURN
820 REM *****
830 REM ** RESPUESTA INCORRECTA **
840 REM *****
850 PRINT
860 PRINT "      ¡INCORRECTO!"
870 FOR Q=1 TO 500:NEXT Q
880 LET B=B+1
890 RETURN
900 REM *****
910 REM ** RESULTADOS EXAMEN **
920 REM *****
930 PRINT TAB(5);"      RESULTADOS DEL EXAMEN"
940 PRINT TAB(5);"_____":PRINT:PRINT
950 PRINT TAB(5);"      RESPUESTAS CORRECTAS ";A:PRINT:PRINT
960 PRINT TAB(5);"      RESPUESTAS INCORRECTAS ";B:PRINT:PRINT
970 IF A>=5 THEN GOTO 1020
980 PRINT TAB(5);"*****"
990 PRINT TAB(5);"***      ¡DEBES ESTUDIAR MAS!      ***"
1000 PRINT TAB(5);"*****"
1010 GOTO 1040
1020 IF A>8 THEN PRINT TAB(5);"ERES UN BUEN ESTUDIANTE.      ¡SOBRESALIENTE!":GOTO
1040
1030 PRINT TAB(5);"      ¡HAS APROBADO!"
1040 INPUT "¿QUIERES HACER OTRO EXAMEN?";Z$
1050 IF Z$="S" OR Z$="s" THEN GOTO 430
1060 END
1070 DATA ¿EN QUE CIVILIZACION SE DAN LAS PRIMERAS FIGURAS DESTACADAS DE LA FILO
SOFIA?,¿CUAL ES LA PRIMERA PREGUNTA REALMENTE FILOSOFICA QUE SE HACE EL HOMBRE?,
¿A QUE SE REFIERE EL PERIODO ANTROPOLOGICO?
1080 DATA ¿QUIEN ESCRIBIO LA "ETICA A NICOMACO"?¿QUIEN ESCRIBIO EL "LIBRO DE LA
REPUBLICA"?¿QUIEN UTILIZO LA PALABRA HOMEOMELIAS?,¿CON QUE PERIODO DE LA FILOS
OFIA SE CORRESPONDE EL ESPLENDOR DEL MUNDO GRIEGO?
1090 DATA ¿DE QUIEN SON LAS LLAMADAS 5 VIAS?,¿QUE CORRIENTE DEFIENDE LA CONSUSTA
NCIALIDAD ENTRE CUERPO Y ALMA?,¿A QUE CORRIENTE DEL PENSAMIENTO PERTENECE LA FRA
SE "VIVIR CONFORME A LA NATURALEZA"?
1100 DATA SENECA ERA,¿QUIEN TRATA POR PRIMERA VEZ EL TEMA DE MOVIMIENTO DE CAIDA
LIBRE?,¿QUE AUTOR PARTE EN SU FILOSOFIA DE LA DUDA GENERAL?,¿QUIEN ES EL QUE DE
FIENDE QUE TODO LO PENSABLE ES POSIBLE?
1110 DATA AL HABLAR DEL EMPIRISMO ¿QUE VALOR RESALTAMOS MAS?,HABLANDO DE ILUSTRA
CION ¿A QUE NOS REFERIMOS MAS EXACTAMENTE?,¿A QUIEN CORRESPONDE EL TERMINO "ARJE
"?
1120 DATA ¿A QUIEN CORRESPONDE EL TERMINO "APEIRON"?¿QUIEN ES EL AUTOR DE LA FR
ASE "SOLO SE QUE NO SE NADA"?¿QUIEN ES EL AUTOR DE LA FRASE "PIENSO LUEGO EXIST
O"?¿QUIEN AFIRMO QUE LA VIRTUD ESTA EN EL TERMINO MEDIO?
1130 DATA ¿QUIEN AFIRMO QUE EL HOMBRE ES UN LOBO PARA EL HOMBRE?,¿A QUE CORRIENT
E PERTENECE MARX?,¿QUIEN ESCRIBIO "LA REBELION DE LAS MASAS"?¿QUIEN ESCRIBIO "A

```

APRENDER CON EL ORDENADOR

SI HABLO ZARATRUSTA?

1140 DATA 1 GRECIA,2 ROMA, 3 MESOPOTAMIA,1 ¿QUE ES LA VIDA?,2 ¿COMO SE MUEVE EL MUNDO?,3 ¿QUE HAY EN EL FONDO DE LAS COSAS?,1 AL AGUA,2 AL HOMBRE,3 AL COSMOS
1150 DATA 1 PLATON,2 SOCRATES,3 ARISTOTELES,1 PLATON, 2 ARISTOTELES,3 SOCRATES,1 ARISTOTELES, 2 SANTO TOMAS DE AQUINO,3 SAN AGUSTIN
1160 DATA 1 ANTROPOLOGICO,2 COSMOLOGICO,3 DE LA ETICA,1 ARISTOTELES,2 SAN AGUSTIN,3 SANTO TOMAS DE AQUINO
1170 DATA 1 CRISTIANA,2 ARISTOTELICA,3 NEOPLATONICA,1 CINICOS,2 HEDONISTAS,3 ESCÉPTICOS,1 ESTOICO,2 ESCÉPTICO,3 EPICURO,1 COPERNICO,2 GALILEO,3 NEWTON
1180 DATA 1 KANT,2 DESCARTES,3 ESPINOSA,1 LEIBNIZ,2 KANT,3 SPINOZA,1 RAZON,2 EXPERIENCIA,3 INTUICION,1 MOVIMIENTO CULTURAL,2 MOVIMIENTO POLITICO,3 MOVIMIENTO SOCIAL
1190 DATA 1 THALES DE MILETO,2 PITAGORAS,3 ARISTOTELES,1 PLATON,2 ANAXIMANDRO,3 ANAXIMENES,1 PLATON,2 SOCRATES,3 EPICURO,1 LOCKE, 2 HOBBS,3 DESCARTES,1 ARISTOTELES,2 SENECA,3 GUILLERMO DE OCKHAM
1200 DATA 1 LOCKE,2 HOBBS,3 ROUSSEAU,1 HUMANISMO,2 RACIOVITALISMO,3 EMPIRISMO,1 MARX,2 HEGEL,3 ORTEGA,1 NIEZTSCHE,2 MARX,3 ENGELS
1210 DATA 1,3,2,3,1,1,1,3,2,1,1,2,2,1,2,1,1,2,2,3,1,2,1,3,1

Modificaciones para otros equipos

El programa es válido para el IBMPC. En los demás equipos habría que efectuar las siguientes modificaciones:

SPECTRUM

50 DIM A\$(25,80):DIM B\$(75,40)
420 PAUSE 0
790 PAUSE 100
870 PAUSE 100
1060 GOTO 9999

Además habría que eliminar la línea 470.

COMMODORE

40 PRINT CHR\$(147)
480 LET W=INT(RND(1)*25)+1
490 PRINT CHR\$(147)
710 PRINT CHR\$(147)

Además habría que eliminar la línea 470.

MSX

480 LET W=LNT(RND(1)*25)+1

Además habría que eliminar la línea 470.

AMSTRAD

Habría que eliminar la línea 470.

PARA LOS MAS JOVENES

Números romanos

Dentro de la gran variedad de numeraciones que existen, la romana tiene mayor importancia entre nosotros, ya que es de las más antiguas conocidas debido a la expansión de esta cultura por el mundo occidental, y que ha llegado a nuestros días tal cual se utilizaba en los años anteriores a Jesucristo. Incluso en muchos casos fue utilizada por los colonizadores de América para aumentar su capital sin que nadie entendiera la complejidad de este sistema.

El sistema de numeración romano está basado en la adición y la sustracción, mediante un número finito de símbolos y siguiendo unas pequeñas reglas; con todo esto es posible representar todos los números naturales.

En Europa fue sustituida por la numeración arábiga y sánscrita hacia el siglo XIII, que ofrecían un uso más fácil y simplificado de la numeración; sin embargo, su uso no se perdió completamente ya que aún en nuestros días los números romanos se siguen usando en cosas tales como la denominación de siglos, de reyes y papas que tienen el mismo nombre y también en el programa que a continuación les presentamos, que convierte números naturales en romanos.

```
10 CLS
20 REM *****
30 REM **      NUMEROS ROMANOS      **
40 REM *****
50 PRINT "*****NUMEROS ROMANOS*****"
*****"
60 INPUT "INTRODUCE NUMERO:";N
```

```

70 IF N<0 THEN PRINT "NUMERO INCORRECTO":FOR X=1 TO 500:NEXT X:CLS:GOTO 60
80 PRINT:PRINT:PRINT:PRINT "EL NUMERO ROMANO CORRESPONDIENTE AL ";N;" ES ";
90 LET A=INT(N/1000)
100 IF A<>0 THEN GOSUB 540
110 LET N=N-(1000*A)
120 IF N>=900 THEN GOTO 160
130 LET B=INT(N/500)
140 IF B<>0 THEN PRINT "D";
150 GOTO 190
160 PRINT "CM";
170 LET N=N-900
180 GOTO 280
190 LET N=N-(500*B)
200 IF N>=400 THEN GOTO 240
210 LET C=INT(N/100)
220 IF C<>0 THEN GOSUB 580
230 GOTO 270
240 PRINT "CD";
250 LET N=N-400
260 GOTO 280
270 LET N=N-(100*C)
280 IF N>=90 THEN GOTO 320
290 LET D=INT(N/50)
300 IF D<>0 THEN PRINT "L";
310 GOTO 350
320 PRINT "XC";
330 LET N=N-90
340 GOTO 440
350 LET N=N-(50*D)
360 IF N>=40 THEN GOTO 400
370 LET E=INT(N/10)
380 IF E<>0 THEN GOSUB 620
390 GOTO 430
400 PRINT "XL";
410 LET N=N-40
420 GOTO 440
430 LET N=N-(10*E)
440 IF N=9 THEN PRINT "IX";:GOTO 660
450 LET F=INT(N/5)
460 IF F<>0 THEN PRINT "V";
470 LET N=N-(5*F)
480 IF N=4 THEN PRINT "IV";:GOTO 660
490 IF N=0 THEN GOTO 660
500 FOR I=1 TO N
510 PRINT "I";
520 NEXT I
530 GOTO 660
540 FOR I=1 TO A
550 PRINT "M";
560 NEXT I
570 RETURN
580 FOR I=1 TO C
590 PRINT "C";
600 NEXT I
610 RETURN
620 FOR I=1 TO E
630 PRINT "X";
640 NEXT I
650 RETURN
660 PRINT:PRINT:PRINT
670 INPUT "OTRO NUMERO (S/N)";A$
680 IF A$="S" OR A$="s" THEN GOTO 10
690 END

```

Modificaciones para otros equipos

Este programa es válido para los ordenadores IBM PC, MSX y AMSTRAD.

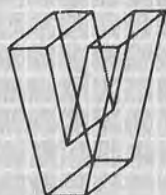
Variaciones para el ZX-SPECTRUM

690 GOTO 9999

Variaciones para COMMODORE

10 PRINT CHR\$(147)

PEQUEÑA HISTORIA DE LA INFORMATICA



EAMOS ahora el valle desde otra perspectiva, no demasiado altruista: la del dinero. ¿De dónde provenía el capital que invertían los «novatos» en sus empresas? ¿De bancos convencionales? ¿Eran ellos mismos ricos?

¿Tenían acaso tal «gancho» como para convencer a financieros, o se asociaban con ellos? De todas estas posibilidades, la más común es la de financieros aventureros, que invierten su capital en empresas porque creen en su futuro. El riesgo que corren es enorme, pero las ganancias que pueden obtener también son muy sustanciosas. Normalmente, este tipo de personas no sólo invierte el dinero en la empresa o negocio incipiente, sino que suele asesorar y casi «sufrir» con el empresario sus avatares empresariales. Así, por ejemplo, le pueden ayudar indicándole asesores fiscales, empresas de publicidad, aconsejándole en problemas de personal, etc. También suelen asesorarle indicándole errores comunes que no debe cometer, etc. En suma, el «financiero» de estas empresas incipientes suele ser, o al menos pasa a ser, un «amigo-consejero» del empresario. Ambos saben que se encuentran en el mismo barco sorteando tempestades. Sin embargo, normalmente los financieros no dependen de una única empresa. Para ellos no es su único modo de vida, y en algunos casos pueden asfixiar y retrasar los planes de ésta, por no comprender aspectos técnicos o empresariales y estudiar únicamente los problemas económicos. Los financieros de la zona, hoy en día, tienen unas redes de comunicación inmejorables, y, por supuesto, están mecanizados con la tecnología puntera que financian. Por ello, no es en absoluto ren-

table para cualquier empresario engañar u ocultar algún dato a su socio inversor, ya que repercutiría probablemente en que nadie le concedería crédito a partir de ese momento.

Sin embargo, aunque el riesgo de las inversiones en investigación es enorme, tampoco es tan «a ciegas» como puede parecer en un principio. Los créditos se otorgan siempre «por fases», de forma que el primer pago o crédito obtenido se entrega para comenzar a arrancar el proyecto. Otra fase del crédito se obtiene cuando va a comenzarse la producción, una vez salvados muchos escollos, y finalmente otras cantidades llegan al empresario una vez que el inversor ha comprobado que el riesgo de perder una cantidad importante (suma de las anteriores cantidades) es pequeño. Con este sistema, el problema se reduce mucho, ya que sólo la primera cantidad se basa en la confianza, siendo las restantes fruto del esfuerzo del empresario.

Siempre existen, sin embargo, determinado tipo de personas que se «niegan sistemáticamente» a recibir cualquier tipo de finan-

¿Sabía usted que...

¿Sabía usted que los ordenadores electrónicos se han programado para todo tipo de tareas? Desde escribir poesía hasta cartas de amor, música para cuerda, etc. Un programador buscando pruebas heurísticas sobre geometría clásica, probó que en un triángulo isósceles los ángulos de la base B y C son iguales siempre que los ángulos ABC y ACB sean congruentes. Esta anécdota no tendría interés alguno si no es porque el programador no conocía en absoluto el teorema, aunque llegó a su solución unos años más tarde que Pappus, su descubridor (siglo III antes de Cristo).

PEQUEÑA HISTORIA DE LA INFORMATICA

ciación. En la vida cotidiana, no utilizan tarjetas de crédito, ni pagan jamás nada a plazos. Esta actitud puede ser muy loable a pequeño nivel (aunque tiene sus inconvenientes también), pero para una empresa supone que su crecimiento sea mucho más lento, y en estas empresas electrónicas e informáticas, el factor tiempo es primordial. Al tomar este tipo de decisiones hay que considerar también si es factible, es decir, si la empresa tiene beneficios, y puede dedicarlos a autofinanciarse. Si es así, y los beneficios son suficientes, la decisión puede ser muy acertada, ya que los porcentajes que se llevan los inversores suelen ser muy altos. Por el contrario, en ciertos períodos de crecimiento, la financiación puede ser vital, el ser o no ser de la empresa.

Uno de los financiadores más afincados en el valle del Silicio es Arthur Rock, y no vino del pueblo de al lado. Vino de Nueva York, donde su padre era propietario de una pastelería. Hoy es uno de los inversores de la familia Rockefeller, una de las fortunas más grandes del mundo. Arthur Rock estudió en Harvard, y en 1951 comenzó a operar en Wall Street (naturalmente, con capital ajeno, ya que en esa época no disponía en absoluto de capital propio). Poco a poco fue introduciéndose, participando en algunas transacciones. En 1957, Bob Noyce y otros más que trabajaban con Shockley estaban buscando inversores en su nuevo negocio. Sus dificultades eran enormes. Arthur Rock fue el que los puso en contacto con Fairchild, que en ese momento estaba dedicado principalmente a la fotografía. A partir de ese momento, Rock comienza a interesarse por los negocios de tecnología punta, y decide mudarse a la zona más vanguardista: California, y en concreto, la Bahía de San Francisco.

En 1961 ya estaba asentado en la zona, e iba adquiriendo un cierto renombre de que las personas en las que invertía tenían éxito en sus empresas. Era verdad. Arthur Rock no estudiaba el proyecto cuidadosamente. Además, en sus principios tampoco disponía de asesores que pudieran explicarle las posibilidades. No era necesario saber tecnología, era más importante seleccionar adecuadamente los hombres que iban a sacar adelante el plan. Esos hombres debían ser trabajadores empedernidos, estar entusiasmados con sus ideas, y como antes hemos explicado, ser honrados. Honrados al valorar sus posibilidades con honradez, no sobredimensionándolas, y siendo

flexibles para reconocer los errores y subsanarlos. La soberbia rara vez lleva al éxito, y, desde luego, hace el aprendizaje mucho más lento.

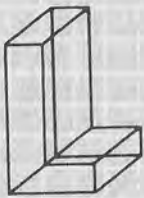
Los financieros que se atreven a apoyar este tipo de negocios son, desde luego, hombres convencidos de la necesidad de investigar, mejorar, innovar. Los riesgos son enormes, y además de tener dinero, es necesario tener vocación. Esa vocación es la que les anima a ayudar en otras labores de asesoramiento a sus «financiados». Además, muchos de ellos dedican sumas a investigación de forma gratuita, destinando fondos a las Universidades, a las que están ligadas, en cierto modo, las empresas a las que financian. Estos fondos permiten subir el nivel de investigación y el prestigio de las Universidades.

Desde principios de siglo han ido apareciendo en la zona sociedades inversoras, cuyo trabajo consiste en reunir capital de distintas fuentes para invertirlo. Algunas han ido adquiriendo solera y prestigio, e incluso tienen publicaciones de importancia nacional. Entre las más conocidas está la American Research and Development Corp., fundada por un general después de la Segunda Guerra Mundial. Otras son la Alex y Brown, Rothschild, Utenberg Townbin, ambas de la otra costa, aunque dedicadas a este tipo de inversiones, la Robertson Colman and Stephens y la Hambrecht and Quist, las dos de San Francisco.

¿Sabía usted que...

¿Sabía que a mediados de los años sesenta, próximas las elecciones presidenciales en las que se presentaba Barry Goldwater frente a Johnson, se «fabricó» la máquina Goldwater, que emulaba las respuestas de ese candidato? El ordenador había sido programado para que respondiera a preguntas sobre la coyuntura del momento, hacía proposiciones sobre las disposiciones gubernamentales, predecía «futuribles», etc. Pero eran los años sesenta. La inteligencia artificial no estaba tan desarrollada como en nuestros días, las máquinas eran menos potentes, etcétera. El caso es que la máquina maravillosa predijo que Fidel Castro «edificaría un muro de Berlín». Como vemos, todavía la informática necesitaba un desarrollo importante para que las respuestas obtenidas de la máquina, aunque excéntricas, al menos no fueran ridículas.

SMALLTALK COMO PARADIGMA DE LOS LENGUAJES ORIENTADOS AL OBJETO*



A programación orientada al objeto se ha desarrollado de la mano del lenguaje Smalltalk: los conceptos básicos de este tipo de programación fueron utilizados por primera vez para describir el Smalltalk, cuando se desarrolló dicho lenguaje y su entorno de programación en el Centro PARC de la XEROX.

Por ello, y a modo de ejemplo de los lenguajes orientados al objeto, presentamos el lenguaje Smalltalk.

El Smalltalk utiliza un pequeño número de conceptos, poco constreñidos por convenciones o términos, al estilo de otros lenguajes de programación, pero muy poderosos. Un procedimiento o subrutina de Smalltalk se llama método. Los nombres de variable en Smalltalk son muy descriptivos, y, por tanto, muy largos. Siguiendo la tradición de Smalltalk, utilizaremos las mayúsculas como separadores visuales, en lugar de utilizar guiones (es decir, se suele escribir *CalculaFrecuencia* en vez de *Calcula-Frecuencias*). Los diseñadores del lenguaje Smalltalk utilizaron un formato para los nombres de procedimiento que anima al programador a describir cada uno de los argumentos. La idea es indicar algo más que el orden de los argumentos, para ayudar al programador a recordar cuál es cuál. Cada parte del nombre de un procedimiento termina por un punto, y va seguido por el argumento que describe.

Por ejemplo, en el programa de las «Torres de Hanoi» (que se presenta más adelante), las cuatro palabras que terminan por puntos, en la primera línea del programa, son la cuatro partes del nombre del método que

está definiéndose. Para referirse al método, junte todas las partes del nombre y refiérase a él como *movetower:from:to:using:*. En este caso hemos utilizado un método con cuatro argumentos. Este método llama a tres métodos (él mismo se llama dos veces).

Digamos otra vez lo que es un objeto en Smalltalk. Un objeto es un paquete de datos y procedimientos que se pertenecen uno al otro. Específicamente, todas las constantes y los contenidos de todas las variables son objetos. Un objeto en Smalltalk es como un registro en Pascal, pero mucho más rico y versátil. Los únicos elementos que no denotan objetos son los selectores de mensajes (operadores o nombres de procedimiento), los comentarios, y unos pocos caracteres de puntuación.

Los diferentes sistemas existentes funcionan de muchas formas: llamando a los procedimientos, aplicando los operadores a los operandos, y, en ciertos casos, ejecutando bloques de un modo condicional. Sin embargo, el Smalltalk sólo tiene una forma de realizar su tarea, y ésta es «enviando mensajes» a los objetos. Un «mensaje», es un selector de mensajes con sus operandos. El objeto que recibe el mensaje, el receptor, aparece justo a la izquierda del mensaje. El Smalltalk siempre devuelve un valor como resultado de cada método, y como era de esperar, el resultado también es un objeto. Los nombres objeto y método vienen de la respuesta que dio uno de sus creadores a la pregunta ¿Cómo seleccionaremos el método por el cual un objeto responderá al mensaje? La respuesta que dio es «Utilice el selector para encontrar el método adecuado que va a ejecutar».

Por ejemplo, la frase «altura > 0» redactada en Smalltalk se escribiría no como «altura mayor que cero», sino como «El objeto altura recibe el mensaje «más alto que» con el argumento cero».

* Como conclusión del tema, aparecido anteriormente en los «Lenguajes orientados al objeto», se describe en este tomo el Smalltalk.

TEMAS MONOGRAFICOS DE VANGUARDIA

Cuando un objeto recibe un mensaje, busca el nombre del mensaje para ver si lo comprende. Si lo encuentra, comienza a ejecutar el método que le indica cómo responder al mensaje.

Del mismo modo que un procedimiento en Pascal puede llamar a otros procedimientos, un método puede requerir llamar a otros métodos. La forma de arrancar otro método es enviar un mensaje al objeto. En ocasiones puede que usted desee enviar un mensaje al mismo objeto que recibió el mensaje en ese momento. ¿Cómo se llamará localmente a ese objeto? En otras palabras, cuando un objeto en Smalltalk habla consigo mismo, ¿cómo se llama? Por supuesto, «self» (él mismo). Los mensajes a *self* son muy corrientes. Cuando una parte de programa envía un mensaje al mismo selector que el método que se está ejecutando en este momento, el programa está utilizando recursión.

La especificación de un objeto, el envío de un mensaje y la obtención de otro como resultado son las únicas cosas que pasan en un programa escrito en Smalltalk. El control de estructura y los operadores aritméticos son simplemente mensajes enviados. El resultado de un mensaje se puede utilizar como objeto que recibe otro mensaje o como argumento en otro mensaje.

Los entornos de programación en Smalltalk intentan proporcionar todas las herramientas necesarias para buscar, ver, escribir y ejecutar los métodos de Smalltalk. El sistema nos puede decir que un pedazo de texto en particular es un método, por la ventana en la que está escrito. Así, no es necesaria una puntuación especial para marcar el comienzo o el final en un método de Smalltalk.

En los sistemas de programación tradicionales, los nuevos programas se crean uniéndolos a otros programas a través del sistema operativo. En Smalltalk, por el contrario, cada programa es una pieza del sistema completo, y todas las piezas están unidas.

El overhead (exceso de trabajo creado) de llamar a un procedimiento de Smalltalk es pequeño, tanto para el sistema como para el programador. El sistema tiene unas once llamadas de procedimiento optimizadas porque ocurren con una enorme frecuencia (típicamente de diez a cincuenta veces por cada procedimiento). El overhead para un programador cuando crea un nuevo procedimiento de Smalltalk es también pequeño. Las declaracio-

nes y preámbulos son muy cortos en el sistema Smalltalk-80, y se puede editar, ejecutar y depurar sin cambiar los entornos. En Smalltalk compilar un método y unirlo (linkarlo) dentro del sistema sólo toma unos pocos segundos, en contraposición a los procesos batch que consumen varios minutos, ya que se necesita recompilar muchos módulos. La escritura de un nuevo procedimiento y la llamada desde el programa original es más fácil en Smalltalk que en Pascal. De este modo, los programadores se ven animados a crear módulos de código nuevos cada vez hay una necesidad nueva.

Para ver la ventaja de programar en Smalltalk, incluimos un programa que calcula el número de veces que aparece cada carácter alfabético en una línea de texto que facilita el usuario. Se dan tres versiones: un programa en PASCAL, su «transcripción» en Smalltalk (véase cómo pueden ser muy parecidos) y una versión optimizada en Smalltalk.

```
const
tamaño = 80;
var
s: string [tamaño];
i: integer;
c: char;
f: array[1..26]
  of integer;
k: integer;
begin
writeln('introduzca el texto');
readln(s);

for i := 1 to 26 do
  f[i] := 0;
for i := 1 to tamaño do
  begin
    c :=
      asLowerCase(s[i]);
    if isLetter(c) then
      begin
        k := ord(c)
          - ord('a')
          + 1;
        f[k] := f[k] + 1
      end
    end;
  end;
for i := 1 to 26 do
  write(f[i], ' ')
end
```

A. Programa en Pascal

```

| s c f k |

f := Array new: 26.

s := Prompter
  prompt: 'introduzca el texto'
  default: ".
1 to: 26 do: [ :i |
  f at: i put: 0].
1 to: s size do: [ :i |

  c := (s at: i) asLowerCase.

  c isLetter
    ifTrue: [
      k := c asciiValue
        - $a asciiValue
        + 1.
      f at: k put: (f at: k) + 1
    ]
  ].

^ f

```

B. Transcripción en Smalltalk.

```

program hanoi(input, output);
var total : integer;
procedure movetower (Height, Frompin,
  Topin, Usingpin : integer);
procedure movedisk ( Frompin,
  Topin : integer);

begin
  writeln( Frompin, '->', Topin)
end, { movedisk }
begin { movetower }
  inf Height > 0 then
  begin
    movetower(Height-1, Frompin,
      Usingpin, Topin);
    modevisk( Frompin, Topin);
    movetower (Height-1, Usingpin,
      Topin, Frompin)

  end
end; { movetower }
begin { hanoi }
  read(total);
  movetower(tota,1,3,2)
end. { hanoi }

```

Versión en Pascal.

```

| s f |
s := Prompter prompt: 'introduzca el texto' default:".
f := Bag new
s do: [ :c | c isLetter ifTrue: [f add: c asLowerCase]].
^f

```

C. Versión Smalltalk optimizada.

```

moveTower: height from: fromPin to: toPin
  using: usingPin
(height > 0) ifTrue: [
  self moveTower:(height - 1) from:
    fromPin to: usingPin using: toPin.
  self moveDisk: fromPin to: toPin.
  self moveTower: (height - 1) from:
    usingPin to: toPin using: fromPin]

```

Versión en Smalltalk que utiliza el siguiente procedimiento:

```

moveDisk: fromPin to: toPin
Transcript cr.
Transcript show: (fromPin
  printString, '->', toPin printString).

```

Fig. 1. A, B y C.—Programa que calcula el número de veces que aparece un carácter en una línea de texto facilitada por el usuario.

Por último, incluimos las versiones en Pascal y Smalltalk de otro conocido problema: las Torres de Hanoi.

Fig. 2.—Dos versiones del conocido programa las «Torres de Hanoi».

■ Los ordenadores y la física de las partículas

La Física de Alta Energía es una rama de la ciencia cuyas raíces llegan hasta los experimentos de Cockroft y Walton. En 1932, Cockroft y su equipo utilizaron un rayo de protones acelerados para dividir el núcleo del litio en dos núcleos de helio. De este modo quedó demostrado que los núcleos atómicos eran divisibles. Desde aquellos días, en los que los cálculos de los resultados experimentales se podían llevar a cabo mediante reglas de cálculo, el desarrollo del equipo experimental ha sido realmente impensable. Este desarrollo, naturalmente ha sido parejo al progreso experimentado en las técnicas de computación, empleadas en la evaluación de los datos experimentales, y ha ido creciendo paralelamente con él. El impacto de los ordenadores en todas las áreas de la física experimental de alta energía, ha sido trascendental para esta Ciencia.

Este campo de experimentación comparte una característica común con los ordenadores: su dependencia del progreso de la electrónica le permite incrementar enormemente sus posibilidades y llevar a la realización práctica otras ideas y posibilidades. Con el cambio de los tubos de vacío a los transistores, así como la integración, cada vez más acusada, los circuitos electrónicos han ido haciéndose poco a poco más pequeños, rápidos y con un menor consumo de potencia. Estas técnicas han permitido la construcción de ordenadores de potencia modesta, pero también de grandes ordenadores de potencial inmenso. Al mismo tiempo, el equipamiento de la física de alta energía, también basado en la microelectrónica en estos últimos 20 años se ha complicado enormemente, de forma que el tamaño de los aceleradores y detectores es ahora tal que los primeros no parecen tener mucho que ver con los segundos.

Examinaremos este impacto de la utilización de los ordenadores en la investigación de la física de alta energía. Para comenzar, describiremos los dos componentes básicos del equipo necesario para la experimentación de este tipo: los aceleradores, utilizados para obtener los rayos de partículas necesarios para iniciar las interacciones a investigar, y los detectores experimentales que registran los fragmentos resultado de las colisiones de las partículas del rayo con algún otro objeto.

La dependencia del diseño y operación de estos componentes con los ordenadores es cada día más estrecha.

Intentaremos dar una somera idea del sistema que se usa en experimentación, utilizando un experimento modelo. El objetivo es no sólo explicar cómo la física de alta energía necesita la informática de forma vital para avanzar, sino también cómo ha influido en el desarrollo de esta ciencia.

■ Física de las partículas

Veamos someramente algunos de los objetivos de la experimentación de la Física de Alta Energía.

Desde los primeros tiempos, el hombre ha intentado comprender el mundo que le rodeaba. En nuestra vida cotidiana, experimentamos dos aspectos de este mundo físico, la materia con la que está formado, y las fuerzas que actúan sobre esta materia. Los antiguos griegos consideraban que la materia estaba constituida por cuatro elementos básicos, el aire, el agua, la tierra y el fuego, y hasta la era de la química moderna, no existió un conocimiento serio de la naturaleza de la materia. Las fuerzas de la naturaleza se manifestaban en muchos fenómenos diversos, como, por ejemplo, los imanes permanentes, la producción de luz y el movimiento de los cuerpos celestiales.

La física de alta energía estudia la naturaleza de la materia y las fuerzas que sólo actúan sobre ella. También se le llama Física de las Partículas, ya que su estudio conlleva una investigación sobre las partículas de materia y las fuerzas que interactúan. El término de alta energía deriva de que la investigación suele tener que llevarse a cabo por medios semejantes a los utilizados por Cockroft y Walton, es decir, mediante rayos de partículas. Los aceleradores modernos proporcionan a las partículas individuales de dichos rayos enormes cantidades de energía, generalmente equivalentes a varios cientos de veces la masa en reposo de la misma partícula.

Las partículas constituyentes de la materia pertenecen a dos clases diferentes: hadrones y leptones. Los hadrones tienen tamaño y masa finitos, y están compuestos de pares o tripletes de otras partículas fundamentales, o cuarks. Un ejemplo de hadrón es el protón, constituyente del núcleo del elemento más sencillo de la naturaleza, el hidrógeno.

Como todos los hadrones, el protón está formado por una combinación de dos tipos de cuares. Uno de ellos lleva una carga aproximada de unos dos tercios la carga del electrón, y el otro lleva una un tercio de la carga de contraria. El protón lleva dos cuares del tipo primero y uno del segundo, de forma que los tres cuares juntos tienen una carga de 1 (la carga del electrón de la órbita del hidrógeno). También podemos considerar el pion, compuesto por un par cuar anticuar, con carga positiva, negativa o cero. El pion positivo consiste en un cuar y la anti partícula del cuar de $1/3$, llamada anticuar. Una antipartícula es una partícula que tiene la misma masa que la partícula correspondiente, pero cuya carga y otros atributos tienen signo distinto. Así pues, vemos que la carga de un pión positivo es $2/3 - (-1/3)$, o una unidad. Los hadrones están naturalmente sujetos a fuerzas que gobiernan sus interacciones.

Como todo el mundo sabe, un acelerador de partículas es un dispositivo muy complejo, y su diseño y puesta a punto son tareas de enorme trascendencia. Aunque el acelerador está compuesto por muchas subunidades que se repiten en anillo, de hecho, lleva cientos de magnetoimanes, cables, microprocesadores de control, y las fuentes de alimentación para los componentes activos. Por tanto, el diseño y construcción de un acelerador constituyen una tarea ingente.

Los ordenadores pueden hacer mucho para ayudar a resolver la tarea, en todos los niveles. En primer lugar, pueden utilizarse para el propio diseño de la máquina, de cada uno de sus componentes. También pueden simular el paso de partículas por el acelerador considerando en su totalidad, o por subunidades específicas. Se han hecho muchas aplicaciones a este respecto, desde una programación batch (sin colaboración alguna por parte del operador) hasta otro tipo de programación interactiva, que permite al diseñador colaborar con la máquina, para optimizar los parámetros de cada subunidad. Sin embargo, la experiencia ha demostrado que el número de parámetros es tan grande que sobrepasa la capacidad del diseñador. Por tanto, normalmente se utilizan programas no interactivos escritos en lenguaje FORTRAN (el lenguaje de programación mejor adaptado a las necesidades y formulaciones científicas).

Se han aplicado técnicas de computación con éxito en los campos siguientes:

1. En la optimización de los muchos tipos de electroimanes, campos y espacios entre los elementos activos, donde existen interacciones mutuas entre partículas del rayo que a veces suelen ignorarse.

2. En el diseño de las propias cavidades del acelerador de partículas.

3. En la simulación de ciertos efectos que no tienen soluciones analíticas o matemáticas simples, especialmente ciertos tipos de inestabilidades que pueden desarrollarse en el rayo que viaja, según va afectándole la influencia de los diversos campos.

También se requieren otros programas para calcular los parámetros de cada electroimán individualmente, o para calcular el sistema de transporte que debe utilizarse para transferir los rayos desde un acelerador a otro, o bien para el cálculo del apantallamiento necesario para proteger al personal de los efectos del rayo y de las radiaciones residuales. Este tipo de programas tan específicos se suelen desarrollar en los Laboratorios de Investigación, por un equipo de personas versadas en los experimentos, y a la vez expertos en programación. Sin embargo, es frecuente intercambiar este software a nivel mundial, entre los distintos laboratorios que disponen de aceleradores de partículas.

A otro nivel, el ordenador se puede encargar de los diseños concretos de los distintos componentes del acelerador, creando los diseños de fabricación necesarios, para que las firmas especializadas puedan llevar a cabo el trabajo. También será necesario diseñar muchísimos circuitos. Estos últimos suelen diseñarse utilizando técnicas CAD-CAM. Existen muchos programas muy interesantes de diseño, que aumentan la capacidad del diseñador, y producen diseños fiables. Son programas interactivos (existe un cierto "diálogo" entre la máquina y el operador). Por ejemplo, este tipo de sistemas puede determinar si se puede mover físicamente un componente dado y colocarlo por ejemplo, detrás de otros dados, en el túnel. También se puede determinar rápidamente la posición del centro de gravedad de un detector voluminoso y complejo, de forma que se pueda manipular con seguridad (grúa, etc.).

Los proyectos en este campo son tan complejos y requieren sumas de dinero tan considerables, que cualquier pequeña mejora puede suponer un ahorro de dinero muy estimable. (No olvidemos que los proyectos para

centrales de este tipo tienen dotaciones de varios cientos de millones de dólares, para períodos de puesta a punto de varios años.)

Otro elemento muy importante dentro del proyecto es una base de datos centralizada, a la que se pueda acceder desde los periféricos de los distintos servicios. Después de un período de estudio, los diseñadores han comprobado que las bases de datos relacionales, incluso con ordenadores de tamaño medio, pueden cumplir perfectamente las necesidades. Una de las más utilizadas es ORACLE, en un sistema VAX 11/785. Con ella se pueden llevar a cabo comprobaciones sobre la planificación general, utilizando un PERT (Project Evaluation and Revision Technique-Técnica de Evaluación y Revisión del Proceso). También el ordenador puede controlar otras actividades, como, por ejemplo, el transporte de

los componentes, su diseño, el cableado entre ellos, y el sistema general de control. Otras tareas del ordenador pueden ser el control de pedidos de repuestos, la generación de listas de componentes de los circuitos impresos, y finalmente la documentación. Todas estas actividades se pueden integrar en un sistema de base de datos, considerando además de los componentes, sus interrelaciones. Por ejemplo, se puede generar el esquema de trabajo del equipo de instalación, interrogando la base de datos que genera el plan, y los componentes de ésta.

Con el tiempo, el número de procesos que requieren uso de la base de datos irá aumentando, y con ello, también el número de operadores que la manejan, pudiendo llegar a quince o más.

VOCABULARIO

Escape. Cierta tecla que disponen los ordenadores, cuya función es generalmente la vuelta a la situación anterior.

Esclavo. Dispositivo que es controlado por otro, llamado generalmente "maestro" o dispositivo principal.

Escritura. En Informática, se suele designar con este nombre a la operación de grabación de datos en cualquier tipo de memoria de un ordenador, por lo que es una palabra muy utilizada en casi todos los lenguajes, manuales, etc. La operación inversa, de recuperación de los datos, se llama "lectura".

Estado. Condición en la que se encuentra un sistema. A partir de ella se puede deducir su comportamiento a las distintas señales introducidas. Se utiliza para indicar las condiciones del ordenador, un periférico u otro componente cualquiera. Generalmente los ordenadores van provistos de un registro de estado, que controla ciertas condiciones especiales, como los posibles errores, y operaciones no permitidas.

Estructura de datos. Conjunto de datos dispuestos de una forma determinada (puede ser un vector de números enteros, datos en forma de matriz, etc.).

Esperar (wait). Situación que puede darse cuando el ordenador está pendiente de que se realice alguna acción, externa, o interna (como, por ejemplo, cuando está procesando una instrucción).

ETB (End of Transmission Block). Carácter que marca el final del bloque que se está transmitiendo.

Etiqueta. Conjunto de caracteres utilizados como dirección o referencia, para poder identificar un dato o un archivo. También puede referirse a la anotaciones realizadas externamente en las cubiertas de los sistemas de almacenamiento, con el mismo fin de identificación.

ETX (End of Text). Carácter que marca el final de un determinado texto.

Exceso de tres, código de. Notación decimal expresada en código binario en la cual cada dígito decimal N se representa por el número binario $N + 3$.

Exclusión (OR exclusivo). Operador lógico que tiene la propiedad de que si P y Q son proposiciones variables cualesquiera, P OR exclusivo Q es verdadero si cualquiera de los dos P o Q es verdadero, pero NO si lo son los dos; y es falso si los dos son verdaderos o los dos falsos. La exclusión se representa a veces mediante la combinación de otros símbolos, lógicos (P OR exclusivo Q es igual a $P \oplus Q$ y no $(P \vee Q)$).

Exponente. En potenciación, número al cual debe elevarse la base del sistema de numeración que se esté utilizando. (Indica el número de veces que la base debe multiplicarse por sí misma.)

Expresión. Conjunto de operadores y operandos. Puede ser algebraica (sen $A + 2(x+3y)$) o booleana (A NAND $(C \vee B)$).

Extract. (Ver **Extraer**.)

Extraer. Seleccionar uno o más elementos de un conjunto, según un determinado criterio

(por ejemplo, se pueden extraer determinados dígitos de una palabra, según una instrucción o máscara).

Facsímil. Reproducción de un documento o imagen a distancia, a través de líneas telefónicas u otro sistema de transmisión cualquiera.

Factorial. Operación matemática por la cual un número se multiplica por todos los anteriores a él. El factorial de cero es 1. Naturalmente cumple determinadas reglas matemáticas.

FE (Format Efecto). Carácter de realimentación.

Ferrita. Material que contiene óxido de hierro, utilizado en la construcción de determinados tipos de memoria.

FF (Form Feed). Carácter y tecla (generalmente en las impresoras) que provoca un salto de página.

Fiabilidad. Característica que se establece para indicar el escaso número de probabilidades existente para que se produzca un error. Puede utilizarse tanto para el hardware como para el software.

Fibonacci, búsqueda de. Búsqueda dicotómica por la cual en cada una de las fases, el conjunto original o los subconjuntos formados se van subdividiendo sucesivamente por números cada vez más pequeños de las

Series de Fibonacci. Si el número de elementos del conjunto no es igual al número de Fibonacci, el número de elementos del conjunto se supondrá igual al siguiente número de Fibonacci (por exceso).

Fibonacci, número de. Se refiere a un entero en las Series de Fibonacci.

Fibonacci, series de. Serie de números enteros, en la cual cada entero es igual a la suma de los dos enteros anteriores a él en la serie. Formulada matemáticamente, será $x_i = x_0 + x_1$ donde $x_0 = 0$, $x_1 = 1$.

Fichero. Conjunto de registros ordenados según un determinado criterio. Los registros son las unidades del fichero. El criterio de organización de los registros es una característica fundamental del fichero. Puede ser de acceso directo, secuencial, etc. Los registros pueden contener muchos datos, por lo que los ficheros se suelen almacenar en memorias de masa, como unidades de disco duro o flexible, cintas, cassettes, etc.

Fichero directo. Fichero en el que sus registros son accesibles de forma directa (se puede acceder individualmente a cada registro). Este tipo de ficheros van provistos de un indicador que permite identificar directamente el registro deseado. Este indicador puede identificar un determinado número de pista y de sector, para la localización del registro. (Véase **Fichero secuencial**.)

